Evaluation and Comparison of Internet Firewalls

by

Ibrahim Abdul-Rahman Al-Kaltham

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER ENGINEERING

February, 1998

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Evaluation and Comparison of Internet Firewalls

BY

Ibrahim Abdul-Rahman Al-Kaltham

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

Computer Engineering

Feb. 9, 98

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

COLLEGE OF GRADUATE STUDIES

This thesis is written by Ibrahim Abdul-Rahman Al-Kaltham under the direction of his Thesis Advisor and approved by his Thesis Committee. It has been presented to and accepted by the Dean of the College of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER ENGINEERING**

Thesis Committee

Dr. Khalid Al-Tawil (Chairman)

Dr. Habib Youssef (Co-chairman)

Dr. Khaled Elleithy (Member)

Dr. Khalid Al-Tawil
(Department Chairman)

Dr. Abdallah M. Al-Shehri
(Dean, College of Graduate Studies)

16-2-98
Feb. 9, 98

ACKNOWLEDGMENT

I would like to thank my thesis committee chairman Dr. Khalid Al-Tawil, and Dr. Habib Youssef, the co-chairman, for their continuous guidance, and the committee member Dr. Khaled Ellethy for his support .

Thanks are due to Dr. Khalid Al-Tawil for his support and providing me with the required books and other related sources. As the departments chairman, he also showed a great commitment to provide the necessary computer equipment as well as the available expertise within the department of Computer Engineering that are needed to help me in setting up the test-bed in the communication lab. I also wish to thank Dr. Habib Youssef and Dr. Khaled Ellethy for their additional efforts in supporting this work.

I would like to thank my friends in the collage of Computer Engineering and Information & Computer Sciences, as well as in Saudi ARAMCO for their support and help. My thanks to Mr. Abdul-Rahman A. Bayounis of Saudi ARAMCO's network operating system group, for his help and explanations on the use of firewalls and proxies to achieve a secure Internet connectivity from ARAMCO site at Dhahran. Also, I would like to thank the network administrator Mr. Farooq Ashraf and Mr. Kaleemuldeen (a graduate student of the department of Computer Engineering) who helped me with matters related to LATEX and UNIX. Lastly, but not the least, I wish to thank my family for their patience and endurance during the long hours while I was working on my thesis.

Table Of Contents

List Of Tables

List Of Figures

ABSTRACT

Firewalls provide the best protection for private networks against possible intrusion from external sites connected through the Internet. Firewalls, can be used, also, to control access to the Internet from inside the local private network. In addition, firewalls can be used internally to separate private administrative domains. In this manner, access to sensitive information can be controlled and limited to authorized personal. Moreover, through the use of firewalls, outbound connections can be restricted to authorized staff and acceptable sites, therefore, reducing the cost and security risks .

In this study, commonly used solutions achieved through firewalls were discussed and various firewall types were evaluated to provide a common base for firewall product comparison. Different test-beds were designed and implemented, where, one was used as the backbone for an experimental firewall evaluation and comparison. Two firewall packages were tested using a security analysis tool to evaluate their strengths and weaknesses. This was achieved by setting up a test-bed that consist of two local area networks connected via a router to a third local area network representing an external network on the Internet. The two firewalls to be tested were installed to protect each of the two local area networks against attacks and intrusions from the third network. In addition to the above, a new improved firewall architecture and double firewall protection have been proposed.

الخلاصة

إن حدران الحريق (Firewalls) توفر افضل سبل الحماية لشبكات الكومبيوتر الخاصة ضد الدخول الغير مسموح به من المواقع الخارجية المتصلة عن طريق شبكة الإنترنت (Internet). كما أنه يمكن استخدام حدران الحريق للتحكم في عملية الاتصال بشبكة الإنترنت من داخل شبكات الكومبيوتر الخاصة. وبالإضافة إلى ذلك فإنه يمكن استخدام حدران الحريق داخليا وذلك لفصل الأقسام الإدارية المختلفة. وبهذا الشكل فإنه يمكن التحكم في عملية الوصول للمعلومات الحساسة و قصر ذلك فقط على الأشخاص المصرح لهم. كما أنه عن طريق استخدام حدران الحريق يصبح بالإمكان قصر الاتصالات الخارجية على الأشخاص المصرح لهم بذلك و إلى المواقع المقبولة فقط مما يؤدي إلى خفض التكاليف و المخاطر.

في هذه الدراسة قمنا بمناقشة الحلول المختلفة التي يمكن الحصول عليها عن طريق استخدام حدران الحريق و الأنواع المتعددة من حدران الحريق وذلك لتكون أساسا لمقارنة أنظمة حدران الحريق المختلفة. كما قمنا بتصميم و تنفيذ نماذج اختبار مختلفة واخترنا أحدها للقيام بتجربة تقييم و مقارنة حدران الحريق. في هذه التجربة قمنا بفحص اثنين من أنظمة حدران الحريق باستخدام أداة تحليل أنظمة الأمن و ذلك لتقييم نقاط القوة و الضعف لكل منها. لقد تم إنجاز ذلك بإعداد نموذج اختبار متكون من شبكتي كمبيوتر محلية (LANs) متصلة عن طريق موجه (Router) إلى شبكة كمبيوتر محلية ثالثة تمثل شبكة كمبيوتر خارجية على الإنترنت. ولقد تم تركيب حدارتي الحريق المراد اختبارهما لحماية كل من شبكتي الكمبيوتر المحليتين ضد المخاطر الصادرة من شبكة الكمبيوتر المحلية الثالثة. و بالإضافة إلى ما ذكر أعلاه فقد قمنا بتقديم تصميم (Architecture) مطور لجدار حريق يمكن من الحصول على حماية مضاعفة.

CHAPTER 1

INTRODUCTION

Local users in a campus or a company may require the expertise of other users in the field, who are often able to spot new business trends and other up to date information faster than local users. Also, remote users may need to make decisions based on the same data that is available for those at the headquarter. Remote access network technologies provide remote users with reliable access to their private network. For this purpose, wide area network (WAN) digital-service technologies are increasingly used. The choice of technologies depends on individual corporate needs, Some companies need full-time point-to-point connections for their branch offices, where others need connections for telecommunications or mobile employees.

Nowadays, computerized operations and information systems are becoming indispensable for governments, enterprises, and individuals as well. These different users of computers need to communicate, exchange messages and information. This has lead to the proliferation of a variety of computer networks such as Local Area Networks (LANs), Metropolitan Area Networks (MANs), and Wide Area Networks (WANs). Which in turn has caused more security concerns and required more effective controls where passwords are not considered enough. Recently, with the emergence of the world wide computer-network

(Internet), that connects millions of hosts [1] , security risks have increased and became easier to be caused and more difficult to be guarded against.

The nature of computer networks indicates that they are designed to allow the free flow of information. The chief benefit of networks is that they enable users to share information easily. But easy access to information is also the cause of concern for those in charge of keeping a network free from security threats. With modern network technologies, a user can sit at a workstation in one location and have a process connected to a system in another location with files mounted from a system in a third location. Therefore users are able to do their work just as if all of the systems are in the same site as the computer they are logged-on to. All of this is possible, because of the free flow of data which is necessary to the basic functionality of the network. However on the other hand, the free flow of information is contrary to achieving network security. The overall usability of the network should not be greatly affected by security measures used to keep private information and sensitive data insulated from unauthorized access. The demand for security products to guard private networks from intrusion is on the rise as the number of businesses and government agencies connecting to the Internet continues to increase.

In modern network environments, the Transmission Control Protocol / Internet Protocol (TCP/IP) suite is widely used to interconnect computing facilities. Unfortunately, several security vulnerabilities exist in the TCP specification in addition to other weaknesses in some of its implementations. Therefore, and because of these vulnerabilities, an intruder

might be able to attack TCP-based systems to gain a TCP connection or cause denial of service to other legitimate users [2] . According to an Information Week/Ernst and Young Security Survey, one of every five respondents admitted that, in twelve months time, their corporate networks, had been broken into, or had been tried to be broken into, by intruders via the Internet [3] .

Networks need to be insulated from attacks. But when an attack happens, containment is the top priority. One of the most effective forms of protection is partitioning the network into security domains[4] . Network security starts with workable policy. No security measures will be effective unless we know what we want to protect. Security policy should describe what data and systems to protect, what levels of protection are appropriate, and what hardware and software are needed.

To achieve the goal of securing networks without restricting the flow of information, Internet firewalls are used as points of security guarding private networks from intrusion. Internet firewalls main purpose is to control and audit access to services and provide defense, both from inside and outside a private network. Therefore, a mechanism for selectively permitting or blocking traffic between the Internet and the network being protected is required. For instance traffic can be controlled by [5] :

1. Routers at an IP level, by selectively permitting or denying traffic based on source/destination address or port. In this case, at least a degree of direct IP-level traffic

between the Internet and the protected network must be permitted.

2. Hosts at an application level to force traffic to move out of the protocol layer for more detailed examination Application level firewalls unlike routers, do not have the requirement of direct IP-level traffic, but are less flexible since they require development of specialized application forwarders known as "proxies".

There are three types of computer network securities: (1) Internet security, where computers are connected through a firewall, (2) Inter-Company security, where the connection is made through a tunnel, and (3) Intranet security, where computers are connect through multiple firewalls [6] . Regardless of the network type under consideration, using a firewall does not reduce the need for highly skilled system administration and it should not be an excuse to pay less attention to system administration of the site. But on the contrary, when a firewall is compromised, a poorly administered site would be widely-open to intruders and therefore, damage. Firewalls provides barriers, therefore, more time can be spent on site system administration duties and less time responding to incidents and damages. Also, a communications pathway should exist between system administrators and firewall/site security administrators to alert the site about new security problems, and other security-related information [3] .

1.1 Security Issues

The primary objective of information security is to control access to information. Information, should be created, reviewed, modified, or deleted, only, by properly authorized individuals. Access control imposes some security requirements as shown in Figure 1.1. First, confidentiality of personal proprietary, or other sensitive data should be maintained. Secondly, integrity and accuracy of the stored information and programs that manage it should be sustained. Finally, systems, data, and services should be made available to authorized users access [7, 8] .

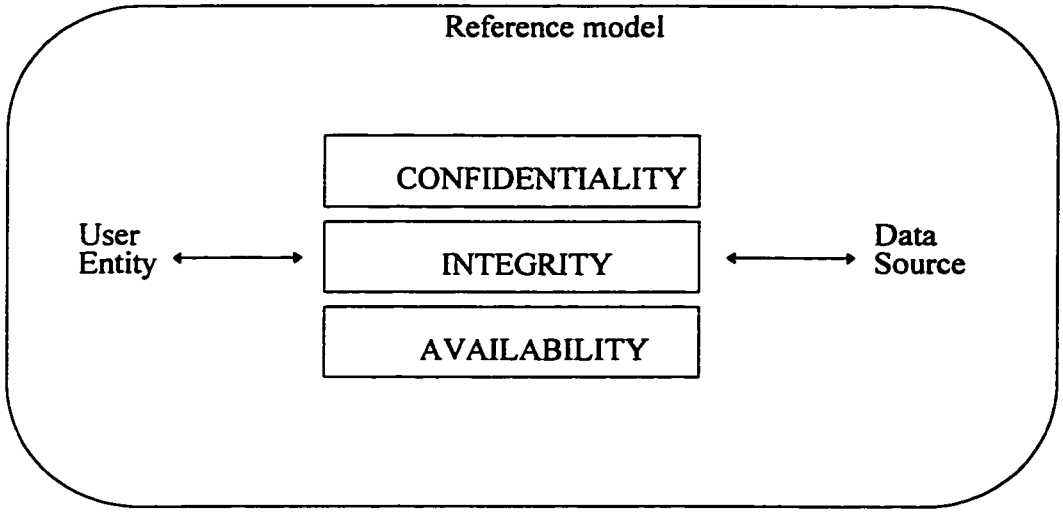


Figure 1.1. Reference Model for Data Management Security and Privacy

Security threats can be classified into two categories: passive attacks and active attacks. In the passive attacks, intruders observe the information passing on without interfering with their flow. Some examples of the class of passive attacks are: picking up electronic emissions from individual computer monitor, or information passing through unprotected

communication lines. While, on the other hand, in active attacks, intruders can modify the information passing on the communication lines. The information can be selectively modified, deleted, inserted, delayed, reordered, or duplicated before reaching its final destination. Some examples of such attacks are: pretending to be another identity, denial of service, and message modification. The potential security threats in heterogeneous networks fall in one of the following categories: unauthorized release of information, unauthorized modification initiation, and unauthorized denial of resources [9] .

Intrusion from both inside and outside the network is on the rise. In the 1994 year-end report by the Ernst & Young auditing and management consulting firm, it was shown that 70 to 80 percent of all computer crimes are carried out by insiders. However, a significant amount of danger is coming onto the network from the outside. Communication over the Internet causes traffic to travel through independent networks controlled by a number of different organizations. The two-way sharing of information offers excellent opportunities. But, at the same time, it produces significant data security challenges. Internet is designed to facilitate communications without any restriction to access computer networks. The very open Internet infrastructure is the cause of it's beauty and problems altogether.

With millions of new connections originating from personal computers and small networks, it is no longer possible to know who or what is on the other end of a network connection unless extraordinary measures are taken. Security is sometimes a mysterious goal and can seem unattainable, especially when we think in terms of the exposure that

an Internet connection offers. But there are workable, practical solutions such as the implementation of firewalls.

Intensive logging may provide information that will eventually lead to the identification of intruders. It is impossible to pinpoint damage, track intruders without detailed audit trails. Firewalls and all sensitive information on the network should be audited, logs should be read and saved for future reference [4] .

Balancing of secure environment need with convenience and accessibility is always difficult. Normally, locks are used to control access by a combination of password authentication and file-access-mode discretionary access control (DAC). Passwords, when machines are relatively isolated, provide a reasonable degree of security. When accessing machines over the network, however, intruders with some effort can either pick the password of an account or intercept it as it transits over the network. Large numbers of network host computers can be cased by an intruder looking for vulnerabilities such as stale or common passwords, alterable system control files, and so forth [10] .

To cope with this increasing threat, several existing packages and facilities have been enhanced. One such package is Kerberos [11] which has been enhanced to plug holes by attempting to discover vulnerabilities before any one else does in order to strengthen existing authentication mechanisms. Also, to avoid passing information in clear text where it might be intercepted, Kerberos-extended network utilities (rlogin, File Transfer Protocol (FTP), and telnet) encrypt communications [10] .

Several solutions have been put together to provide more complete solutions. One of the best solutions to protect computer networks against intrusion and unauthorized access is the use of firewalls. According to a recent study by the Computer Security Institute (CSI) of San Francisco, one out of every five Internet sites has been attacked by an intruder and nearly 40 percent of sites connected to the Net have no firewall in place. As a result, CSI estimates that firewall sales will grow from $1.1 billion in 1995 to $16.2 billion in the year 2000. There are three firewall types: (1) router-based filters, (2) circuit gateways, and (3) application-level gateway [12] .

The Computer Emergency Response Team (CERT) issues advisories that describe security holes in popular products and systems, as well as a complete set of auditing and intrusion testing tools. Security evaluation tools for TCP/IP (Transmission Control Protocol / Internet Protocol) networks are in abundant supply. One of the Security Administrator Tools for analyzing networks is called SATAN. It is designed to detect security vulnerabilities in any computer on the Internet. SATAN can help discover the relative security issues inherent in private networks as well as security faults on networks outside the private administrative domain [4, 13] .

In this thesis we are concerned with the evaluation and comparison of Internet firewalls. To achieve this goal, we implemented one test-bed that we designed for this purpose. A test methodology was defined using this test-bed and SATAN security analysis tool to attack two firewall products (SOCKS v5 and TIS FWTK 2.0). In addition, results of manually

interacting with these two firewalls were observed and reported. Finally, new proposed firewall solutions are presented.

1.2 Thesis Motivation

Firewalls are designed to satisfy different security and functionality goals. They can be of different types and based on different architectures. Therefore, commercial as well as public domain firewall products provide different levels of security and functionality. Normally, these two goals are of conflicting nature.

System administrators need some testing methodology to help them in making the right decision when selecting a firewall to meet their organizations' specific needs. Moreover, administrators need to have a good understanding of security and functionality implications of using a firewall of some type or another. In this thesis, we intend to help system administrators in making the right decision whether to buy or build a firewall. If they decide to buy one, then, what firewall product and of what type it should be.

Whether an organization decides to buy or build its own firewall, it should be implemented using an appropriate architecture. Security and functionality requirements must be selected by the chosen firewall architecture. Therefore, we decided to propose some general purpose firewall solutions that satisfy most requirements, and that are flexible enough to meet cost constraints.

1.3 Thesis Organization

Chapter 2 presents background information necessary to understand security problems that call for firewall protection. It includes the following sections: 2.1) Definitions, 2.2) The Internet and its Security, 2.3) TCP/IP, 2.4) Analysis Tools, 2.5) Protection Methods, 2.6) Security and Policies, and 2.7) Summary.

Chapter 3 discusses firewalls in more details. It includes the following sections: 3.1) Overview, 3.2) Firewall Types, 3.3) Firewall Architectures, 3.4) Firewalls as chapter of the Overall Security Policy, 3.5) Buy or Build a firewall, 3.6) Future Firewalls, and 3.7) Summary.

Chapter 4, Firewall Examples, includes the following sections: 4.1) Digital's Three Way Isolation, 4.2) IpAccess Firewall, 4.3) SURF Firewall, 4.4) TIS Internet Firewall Toolkit (FWTK), 4.5) SOCKS Firewall, and 4.6) Summary.

Chapter 5, Firewall Experimental Evaluation and Comparison, includes the following sections: 5.2) Test-Beds, 5.3) Selection Criteria, 5.4) Setup of The Selected Test-Bed, 5.5) Test Methodology, 5.6) Test Results, and 5.7) Summary.

Chapter 6, Proposed Firewall Solutions, includes the following sections: 6.1) Proposed Firewall Architecture for a Secure Internet Connectivity, and 6.2) Multi Level Firewall Protection, and 6.4) Summary.

In chapter 7, we conclude the thesis and discuss future work.

Finally a number of appendices are included to provide additional information and to show and prove some of the work that has been done.

1.4 Summary

In this chapter, we have discussed network security concerns caused by the emergence of Internet and firewall protection. The problem that security system analysts face when selecting firewalls was formulated. The motivations for taking up this project were also discussed.

CHAPTER 2

BACKGROUND

Engineers often incorporate firewalls into buildings to protect them from possible damages that may be caused by fire. These firewalls serve as fireproof structures to contain the fire within a pre-designated area, or domain. The fire will remain contained as long as the firewall is intact. Things can heat up quickly when flames penetrate the firewall or when its continuity is interrupted by a door. As long as the door is kept closed, it may be effective, otherwise, the firewall will be useless. The same is true with network firewalls [4] .

2.1 Definitions

Definitions for some of the commonly used terms in the terminology of firewalls and computer-network communications are as follow:

Access Path: An access path defines the connectivity path between a user and a computer. Users access information or services from computers via access paths. It can be done through a direct access to the computer, an attached line, or a network connection [10] .

Address translation: Address translation provides a higher level of security by completely hiding the internal network from the Internet. This way nothing will be addressed directly to any internal network device. It is also useful for accessing Internet without using

approved IP addresses. The network address translation feature relieves companies from having to convert all existing addresses to approved ones [14] .

Bastion host: A computer system that is used as a main point of contact to the Internet for users of the internal network. It gets its name from the highly protected overhangs on the outer walls of medieval castles. It must be highly secured because of its vulnerability to attacks, due to its exposure to the Internet.

Bomb: A software bomb is an embedded logic in a program to check for certain conditions on the system. Upon the presence of the checked for conditions, the software bomb executes some function resulting in unauthorized actions.

Dual-homed host: A general-purpose computer system that has at least two network interfaces (or homes)

Encapsulation: The technique used by layered protocols in which each layer adds header information to the protocol data unit (PDU) of the layer above it.

Gateway: In modern usage, the terms "gateway" and "application gateway" refer to systems that perform translation from one format to another, such as electronic mail gateways.

Inetd: Inetd is a UNIX daemon that is responsible for starting other service daemons.

Packet filtering: Sometimes known as screening, it is the action of selectively controlling the flow of data to and from a network. Packets are allowed or blocked by packet filters, while being routed from one network to another (usually from the Internet to an internal network, and vice versa). To accomplish packet filtering, we set up a set of rules that specify what types of packets are to be allowed to or a particular IP address or port and what types of packets are to be blocked from a particular IP address or port. Packet filtering may occur in a router, in a bridge, or on an individual host.

Perimeter network: Sometimes called De-Militarized Zone (DMZ), it is a network added between a protected network and an external network, in order to provide an additional layer of security.

Proxy: Proxies are stripped-down versions of the original programs. For instance, the standard versions of the Unix sendmail utility have about 20,000 lines of code. A proxy version, such as Trusted Information Systems' smap (sendmail application proxy), contains only about 700 lines, because it doesn't include all the functionality of the standard version. It passes along mail messages only after verifying that they fit within the programmed restrictions.

Proxy server: A program that deals with external servers on behalf of internal clients. Proxy servers relay approved client requests from internal proxy clients to real external servers and relay answers back.

Router: A computer or special device used to create a permanent network or Internet connection. It is responsible for making decisions about which of several paths, the traffic will follow. Routers can compare the source or destination address of the IP, or combinations of specific source and destination addresses, against a user-specified table to perform as router-based filters [15] . Routers, also, perform required format translation from the protocol formats of the source network to those used in the target network. They assume that protocols of transport layer and above are the same in both networks.

Sniffer: A sniffer is a program that gathers all network packets transmitted on a given network. Where, it filters these packets according to some defined criteria and report the doubtful packets to the system's security administrator.

Spoofing: Spoofing is the act of making one computer pretend as a different machine or originating IP address, that the receiving system trusts.

TCPWrapper: TCPWrapper programs rely on a simple, but powerful mechanism, in which the inetd is tricked so that it will run a small wrapper program instead of directly running the desired server program. The wrapper will log the client host name or address and perform some additional checks. When all conditions are satisfied, the wrapper will execute the desired server program and terminate.

Trojan horse: A Trojan horse is an undocumented routine that is secretly embedded within a useful program. This secret routine will be executed upon the execution of the program within which it has been embedded.

Tunnels: Tunneling is the practice of encapsulating a message from one protocol in another to use the facilities of the second protocol to traverse some number of network nodes. The encapsulation is stripped off at the destination point and the original message is reinjected into the network [16] .

Virus: A virus is an embedded code in a program, which usually performs some unwanted functions. It propagates by inserting copies of itself into other programs.

Wrapper: See TCPWrapper.

2.2 The Internet and its Security

The term internet, also known as the information highway [17] , is used to mean any connected set of networks. While, the Internet is defined as the world-wide network of networks that uses TCP/IP suite for communications [3] . The Internet has been described as the gateway to a promising future of information availability, technology implementation, knowledge management, cost containment, and several other good and desirable services.

The Internet is a very powerful tool that can be used for good or evil purposes. The great strengths of the Internet are flexibility and openness. But, therein also lies its vulnerability, since this flexibility can be used to cause harm be negligent or ill-intentioned people.

2.2.1 Internet Services

A large number of services and tools are provided through the Internet and its service providers such as: AT&T, Microsoft, and Compuserve. The principal Internet language interface is the TCP/IP protocol. It is possible via TCP/IP that several different kinds of computers communicate with each other and transmit at the same time different kinds of messages, such as data, audio, and video. Internet users use browsers[1] or some other software on their workstations to organize, translate, and display the information that is received from the Internet. Commonly used tools available on the Internet include:

1. E-Mail using Simple Mail Transfer Protocol (SMTP).
2. USENETS & NEWSGROUPS, to provide information and correspondence between users.
3. File Transfer Protocol (FTP), to enable users to download and upload at low costs between different network systems, large volumes of data, including documents, software, images and voice information.

[1] A browser is an application product, such as the Netscape, using a Graphic User Interface (GUI) to establish a connection with other Internet sites in order to review files and other information. It is used primarily for Web site access.

4. The Internet Index and Retrieval Service that acts as a navigation tool to determine the Internet's size, dimensions and topography.

5. Internet Request Services (IRSs) that include World Wide Web (WWW) and Wide Area Information Services (WAIS).

6. Finger which is a UNIX command used on the Internet to search and identify current Internet users as well as their designated locations.

7. TELNET which is another UNIX command that enables uses to directly log onto remote computer systems. This command has a number of negative security implications.

2.2.2 Strengths and Weaknesses

The Internet like other public networks, usually contain more vulnerabilities than private or stand-alone systems. By following accepted principles, users can help in taking benefit of the strengths and reducing the effect of the weaknesses. Just like other business in today's world, standards have become critical in the Internet. The Internet strengths include:

1. A global groundwork for users and suppliers,

2. Open connectivity,

3. Unlimited access,

4. Larger volumes of data,

5. Free valuable information on several sites.

On the other hand, the weaknesses and vulnerabilities of the Internet include:

1. Less security,
2. Complexity of infrastructure,
3. Lack of global control and administration,
4. Lack of technical understanding,
5. Topology is unstable,
6. Quality of information is variable,
7. Network can be overloaded, especially with video, graphic and audio content,
8. Services might be lost,
9. Information in clear-text might be exposed to unauthorized parties,
10. Risks of virus transmission,
11. Attachment to unreliable networks,
12. Lack of standardized backup and recovery methods,
13. Ease of impersonation or spoofing, which results in the difficulty of user identification.

2.2.3 Importance of Internet Security

The two important keys to success in the dynamic and competitive environment of today's business, are information integrity and availability. Both can be achieved only with established protection controls. The threats of unauthorized information modification, disclosure, and destruction must be firmly countered with the appropriate practices. The old mainframe-centric computing paradigms are replaced by some of the new technologies which have made the Internet more popular.

Strict controls and protection measures have been developed on the mainframe over the years for software and data backup, access control, hardware redundancy. For some old systems, the "mean-time-to failure" is measured in decades. On the other hand, this kind of reliability cannot be provided by the Internet, because usually, it is users who are responsible for their own backups, access control, and redundancy. Since the Internet communication is made across public telephone lines, the type of security that is possible on dedicated mainframe communication lines, can be achieved only through encryption or some other user initiated controls.

Mainframe software can be thoroughly tested in a closed development environment, where the test can be made transparent to the user, before the software is made available for operations in the production environment. On the contrary, in the Internet environment, it is the responsibility of users when downloading software from other computers to protect themselves from viruses, Trojan horses, software bombs, and other codes which don't work

as expected. Therefore to shift from mainframe computing to distributed networking require much more effort from end-users and increased attention to protection techniques.

Hierarchical organizational structures are too expensive to maintain, therefore, flatter organizations with authorized employees working in proficiency centers, are becoming the standard and a critical requirement for competition. This shift is made possible through the Internet which provides the necessary communications and access to information. But, authorized employees on the Internet, must be responsible users, otherwise, any competitive advantages may be lost. Another vital component to successfully secure information, is the active participation of users in the implementation of security controls.

2.3 TCP/IP

In modern network environments, the TCP/IP suite is widely adapted for the interconnection of computing facilities. The TCP/IP protocol suite was developed for use with the Advanced Research Projects Agency (ARPA) network by the U.S. Department of Defense (DOD) in the 1970s [17] . Today, it is a public domain protocol and is the most widely used protocol to connect computers to the Internet. It defines how different computer systems can talk to each other in a uniform manner over communication networks. The TCP/IP protocol suite is a group of protocols implemented at four different layers, as shown in Figure 2.1.

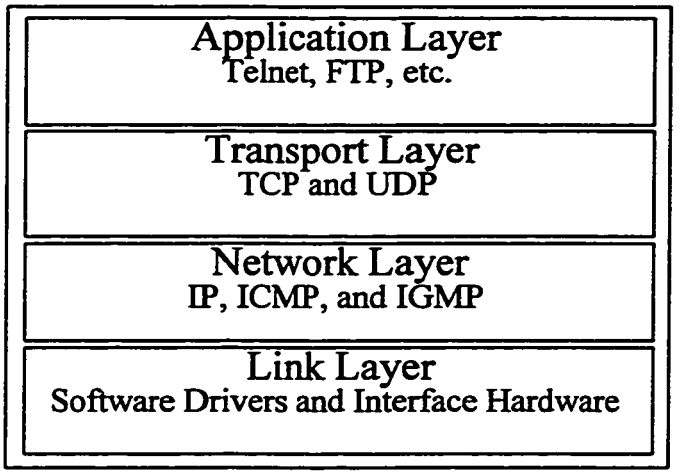


Figure 2.1. The Layers of the TCP/IP Protocol Suite.

The four layers and the protocols that are supported in each layer will be briefly presented [18] .

- The first layer is the Link Layer which describes how the physical interface between the computer and the network is made. Details of device drivers that are used to control the interface card of the communication network physical medium is included at this layer.

- The second layer is the Network Layer. It contains protocols that define how packets are routed around the network. Three of the major protocols that are included in the Network Layer are Internet Protocol (IP), Internet Control Message Protocol (ICMP), and Internet Group Management Protocol (IGMP).

- The third layer is the Transport Layer which realizes the details on the flow of information between two network hosts. In this layer, there are two types of protocols

to provide for either a reliable or non-reliable delivery of data. The reliable delivery of data is achieved via the Transmission Control Protocol (TCP), which guarantees that the information sent from one network host will arrive to the specified destination. On the other hand, non-reliable delivery of data is made through the User Datagram Protocol (UDP), which sends packets from any network host, but it does not guarantee that packets sent will arrive at their destination over the network.

- The fourth and last layer is the Application Layer, where specific details are maintained about running applications on the network host. Some of these applications are: Telnet, File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP).

In the following two subsections, we will discuss in more details the TCP and IP protocols, while more detailed discussion can be found in [19] .

2.3.1 The Internet Protocol (IP)

One of the major protocols in the network layer is the Internet Protocol (IP). The basic unit of information in the IP protocol is called a datagram which consists of header and data parts. As shown in Figure 2.2, the format of the IP datagram header also consists of two parts. The first part is fixed and is of length 20 bytes, while the second one is optional and is of varying length [19] .

The Time to live counter field is used to limit packet lifetimes to prevent datagrams from wandering around forever in case of corrupted routing tables. It is supposed to limit

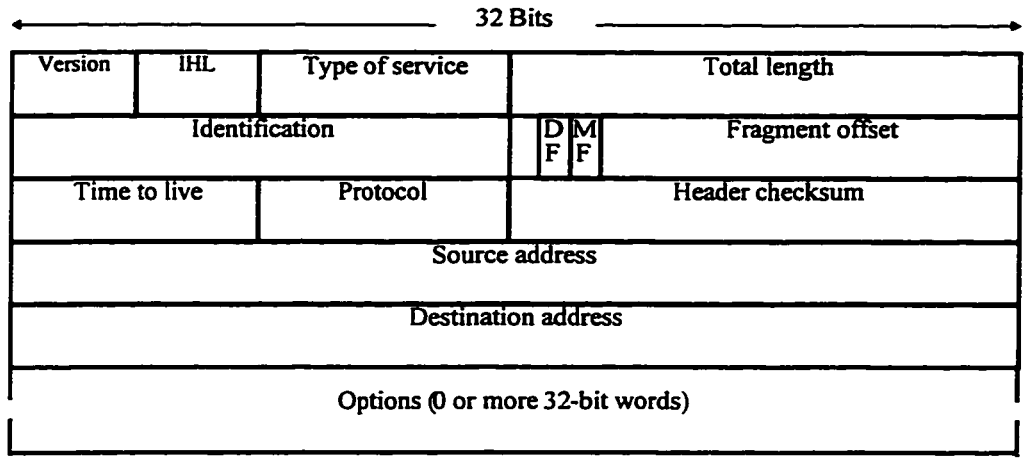


Figure 2.2. The Internet Protocol (IP) header.

the lifetime to a maximum of 255 seconds and be de incremented on each hop as well as when queued for a long time in a router. However, in practice, it only counts hops and when it reaches zero, it is discarded and a warning packet is sent back to the source host.

The Protocol field holds a protocol number that tells the network layer what transport process it should give the completely assembled datagram to. The choice of protocols includes TCP, UDP, and some others. Protocol numbering is standard through out the entire Internet and is defined in RFC 1700.

The Source and Destination address fields indicate the network and host numbers. We will discuss the format of Internet Protocol (IP) addresses, later in this section, when we discuss addressing and subnetting.

The Options field is designed to allow subsequent versions of the protocol to include information not available in the original design. This allows experimenters to try out new

Option	Description
Security	Specifies the secrecy level of the datagram
Strict source routing	Gives the full path to be followed
Loose source routing	Gives a list of routes that can not be missed
Record route	Instructs each router to append its IP address
Timestamp	Instructs each router to append its address and timestamp

Table 2.1. Internent Protocol (IP) options

ideas, while avoiding the allocation of unnecessary header bits for rarely needed information. The options are of variable length, each beginning with a 1-byte identification code and padded out to a multiple of four bytes. Some options are followed by a 1-byte optional length field and one or more data bytes, while maintaining the multiplicity of four bytes. There are five defined options, see Table 2.1, but not all of these five options are supported by all routers.

The Security option identifies the secrecy level of the information in the datagram. It is supposed to be used in routers so that they will not route the datagram through certain untrusted sites. But, in practice, it is ignored, therefore, it only helps spies in finding the valuable information more easily.

The Strict source routing option specifies the complete path from the source to the destination as a sequence of IP addresses. In this case, the datagram is forced to follow this route exactly. The Strict source routing option allows system managers to send emergency packets for timing measurements or when the routing tables are corrupted.

The Loose source routing option forces the packet to pass through the list of specified routers and in the same specified order, while allowing it to pass through other routers on the way. This option is quite useful when it is needed to force packets to pass through or avoid certain sites.

The Record route option instructs routers along the path of the datagram to append their 32-bit IP addresses to the option field. This option can assist system managers in tracking down bugs in the routing algorithms, such as when packets follow a longer path or always visit a certain node.

The last option is the Timestamp option which is similar to the Record route option, but in addition each router will also record a 32-bit time-stamp. This is mostly to debug routing algorithms.

2.3.1.1 Addressing and Subnetting

IP addresses are used in the Source and Destination address fields of IP packets. Every host and router on the Internet has a unique 32 bits long IP address that is composed of network and host numbers as shown in Figure 2.3. Machines that are connected to more than one network have a different IP address corresponding to each network. In order to avoid conflicts, the responsibility of network numbers assignment is delegated to the Network Information Center (NIC). The 32-bit network address number is usually written in dotted

decimal notation, where, each of the 4 bytes is written in a decimal number ranging from 0 to 255. For instance, the hexadecimal address C0290614 can be written as 192.41.6.20 [19].

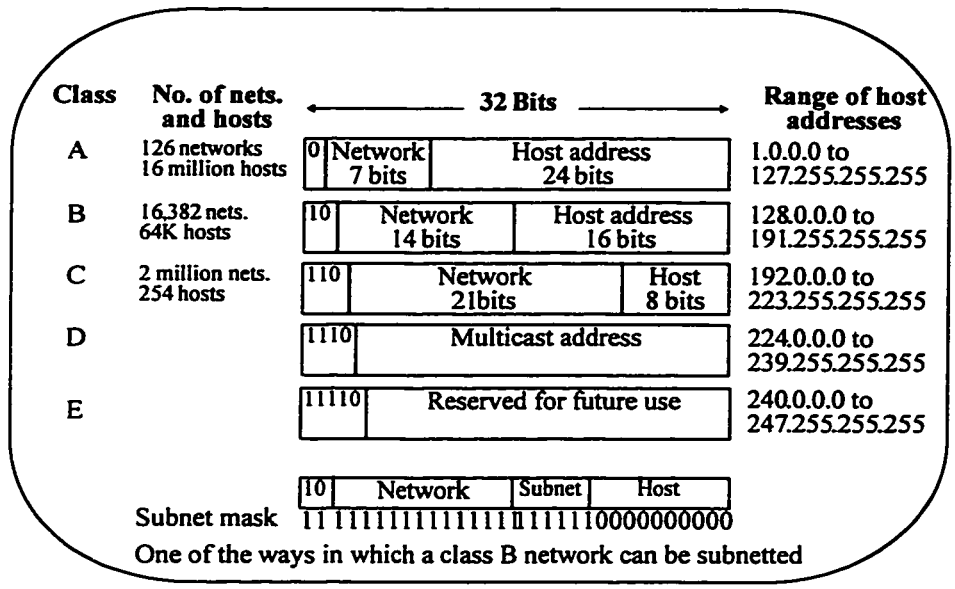


Figure 2.3. The address formats of the Internet Protocol.

As networks grow and more hosts are acquired than what can be supported by the host part of the IP address, the IP addressing property that all hosts in a network must have the same network number can cause problems. When this happens, more network addresses will be needed. Also if an organization acquires a second LAN of a different type and wants a separate IP address for it. One solution is to bridge the LANs form a single IP network, but bridges have their own problems. Therefore, it might end up with many LANs, each with its own router and network number. Management of several distinct local networks is not easy and is not efficient, where system administrators will face problems such as [19] :

1. Every time a new network is installed they will have to contact NIC to get a new network number and announce it worldwide.

2. Moving a machine from one LAN to another involves changing its IP address and announcing it to the world. This in turn may involve modifying the configuration files of the moved machine, and if some other machine is given the released IP address, that new machine will get all data including e-mail intended for the moved machine until the new address has propagated all over the world.

The term subnet is usually used to mean the set of all routers and communication lines in a network. However, this term is used differently here to mean splitting the network into several parts (subnets) for internal use while still operate as a single network to the outside world. The intended meaning can be clear from the context. Subnetting can be the solution to the above mentioned problems. To make this more clear, the 16-bit host number of a class B address, for example, can be split into a 6-bit subnet number and a 10-bit host number, as shown in Figure 2.3. In this way, a number of 62 LANs (0 and 1 are reserved), each with up to 1022 hosts can be obtained. In this example, one subnet might use an IP addresses starting at 130.50.4.1, while another subnet might start at 130.50.8.1, and so on. Since this subnetting is transparent to the outside world, allocating a new subnet does not involve contacting NIC or changing any external application.

When subnetting is used, the routing tables should be changed so that a router on any subnet knows how to get to other subnets as well as to hosts on its own subnet, without

having to know any details of hosts on other subnets. Thus reducing the space required for router tables by creating a three-level hierarchy (network, subnet, and host). All that is needed to be done to perform subnetting, is to have each router perform [19] :

1. ANDing operation of the packet's IP destination address with the network's subnet mask as shown in Figure 2.3, in order to get rid of the host number.
2. Look up of the resulting address in its own tables after determining its network class.

As an example of the above, consider a packet addressed to host 130.50.15.6 and arriving at a router on subnet 5. The arriving packet's IP destination address is then ANDed with the subnet mask of Figure 2.3, resulting in the address 130.50.12.0 which means that the destination host is on subnet 3. The resulting address is then looked up in the routing tables of this router to see how to get to the destination subnet 3.

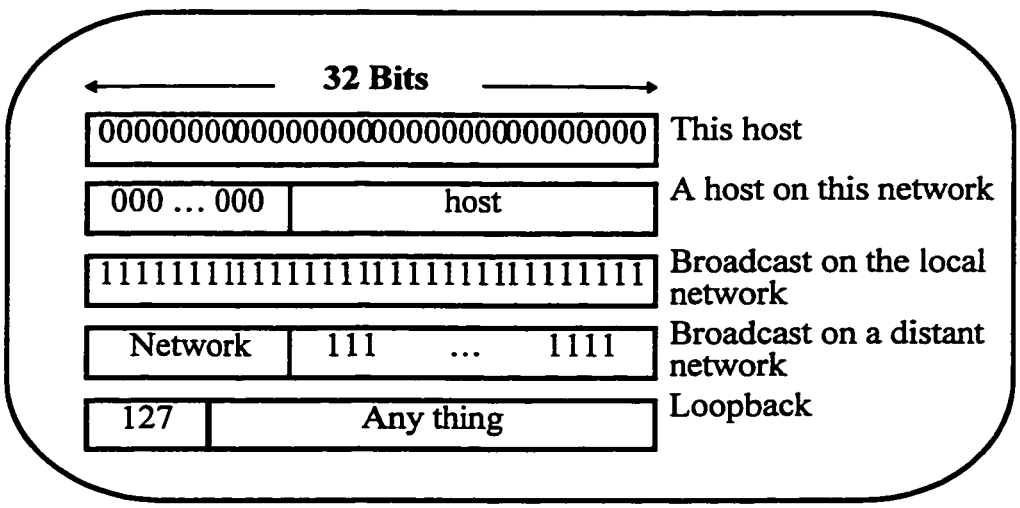


Figure 2.4. Special Internet Protocol (IP) Addresses.

Finally, there are some special IP addresses as shown in Figure 2.4, where 0 and 1 values have been given special meanings. The 0 value is used to mean this network or this host, while the value of 1 indicates all hosts on the specified network for broadcasting purposes.

2.3.2 The Transport Protocol (TCP)

There are two major Internet protocols in the transport layer. The first protocol is the TCP which is a connection-oriented protocol. The second protocol is the UDP which is a connection-less protocol. UDP is basically the same as the IP protocol with the exception of an added short header. For this reason, we will just concentrate on the first protocol, the TCP [19] .

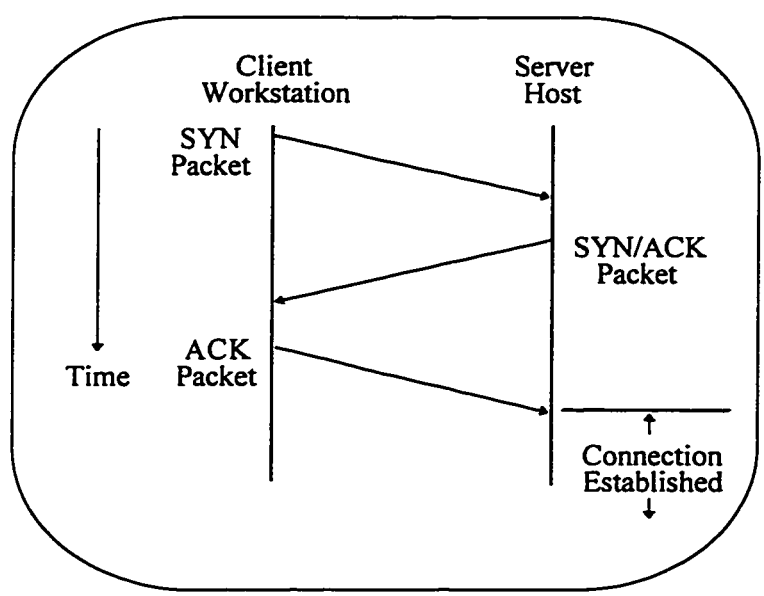


Figure 2.5. Diagram of TCP Three-Way Handshake Connection Establishment.

To establish a network connection between two network hosts using the TCP/IP protocol suite, a three-way handshake takes place as shown in Figure 2.5. At the beginning, a synchronization (SYN) packet containing its initial sequence number is sent from the requesting host to the server host. Then, the server host, responds by sending back a SYN packet with its initial sequence number and an acknowledgment of the requesting host's initial sequence number. And finally, the requesting host acknowledges the initial sequence number of the server, and thus, the TCP connection establishment between the two network hosts is complete [17] .

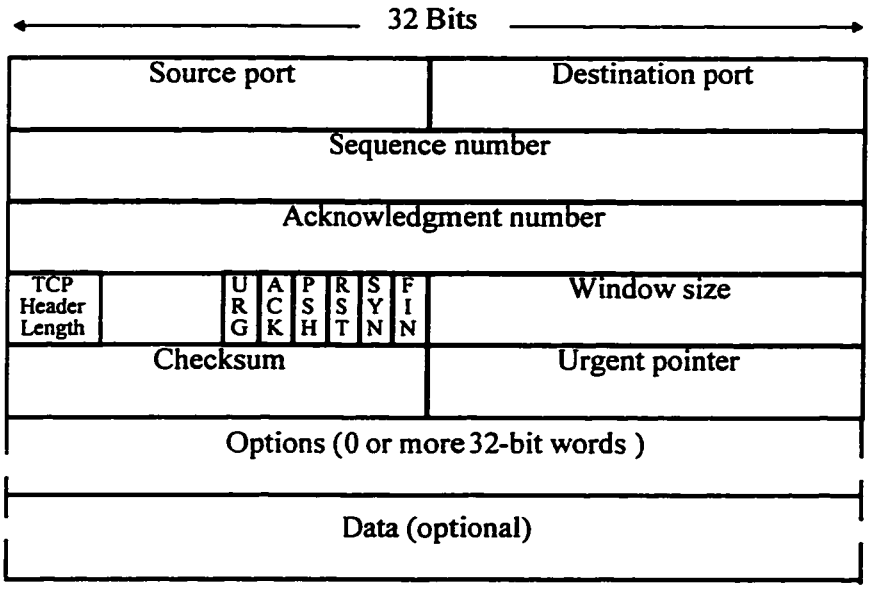


Figure 2.6. The TCP header.

The header of the TCP is 20 bytes long, not counting the variable size section for options that can be of length zero or more of 32-bit words, as shown in Figure 2.6. TCP packets may be broken into several segments, where each segment contains the source

and destination port number to identify the sender and receiver, respectively. To maintain the bytes of data in the proper order of transmission from the sender to the receiver, the sequence number is utilized. The sender and the receiver can differentiate between lost and retransmitted data in the connection, by communicating the sequence number and a corresponding acknowledgment number. The TCP header, contains six flag bits, where one or more of these flag bits can be set at any given time, these flags are: URG, ACK, PSH, RST, SYN and FIN [19] .

- The URG flag bit is set to 1 if the urgent pointer is in use, which indicates a byte offset from the current sequence number at which urgent data are to be found. It allows the sender to signal the receiver for interrupts without involving the TCP itself.
- The ACK flag bit is used to indicate whether the TCP segment contains a valid acknowledgment number or not. If the ACK flag bit is set to 1, it means yes, otherwise, the acknowledgment number is ignored.
- The PSH flag bit means PUSHed data, when this flag is set, the receiver is requested to deliver the data immediately to the application. Otherwise, for efficiency reasons, the received data is buffered until the buffer is full.
- A segment with RST flag bit on, indicates the presence of a problem of some kind. Therefore, the RST flag bit is used to:

* Reset a connection that has become doubtful,
* Reject an invalid segment,
* Refuse an attempt to open a connection.

- The SYN flag bit is used to establish connections. It is used in relation with the ACK flag bit to distinguish between CONNECTION REQUEST and CONNECTION ACCEPTED. In the first case SYN = 1 and ACK = 0, while in the second case SYN = 1 and ACK = 1.
- Finally, the FIN flag bit is used to release a connection. Although, it indicates that the sender has no more data to transmit, after closing the connection, more data may continue to arrive at the receiver. But, since both SYN and FIN segments have sequence numbers, it is guaranteed that the valid segments will be processed in the correct order.

The TCP state-transition diagram governs the initiation, establishment, and termination of a connection. The TCP state-transition diagram consists of well-defined states and transitions between these states. In addition, there are several timers used with the various transitions associated with the connection establishment, connection termination, flow control, and retransmission of data. The various states of the TCP protocol as shown in Figure 2.7 are explained below [19] :

- In the CLOSED state, there is no active or pending connection.

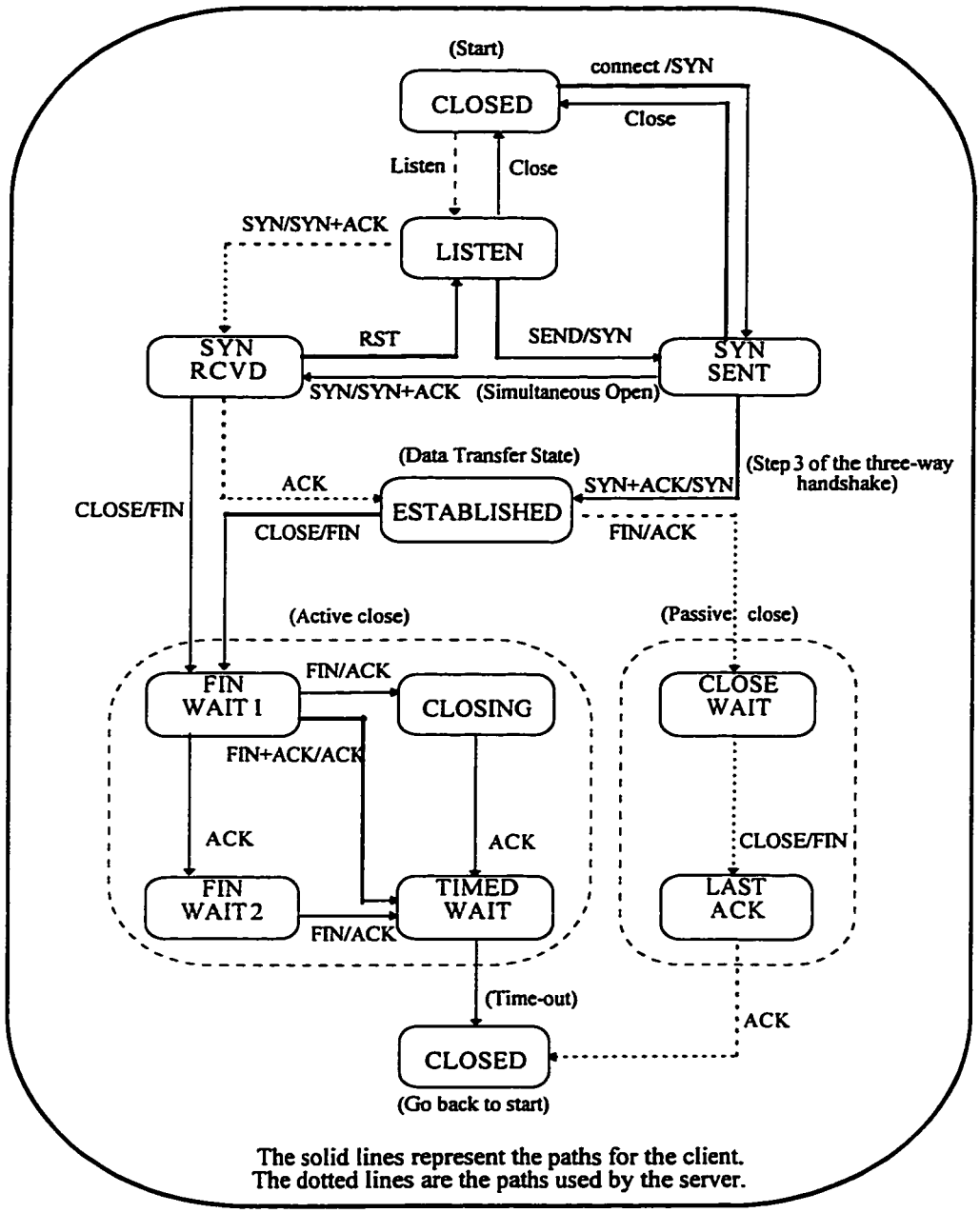


Figure 2.7. The TCP connection management finite state machine.

- When in the LISTEN state, the server is waiting for an incoming call.
- The SYN RCVD state means that a connection request has arrived and is waiting for ACK.
- When an application starts to open a connection, the client moves to the SYN SENT state.
- The normal data transfer state is the ESTABLISHED state.
- FIN WAIT 1 and FIN WAIT 2 states mean that the application has said it is finished and that other side has agreed to release respectively.
- The client waits in the TIMED WAIT state for all packets to die off.
- CLOSING state indicates that both sides have tried to close simultaneously.
- CLOSE WAIT state indicates that the other side has initiated a release.
- Finally, the LAST ACK state means wait for all packets to die off.

2.3.3 Security of the TCP/IP Suite

Even though, the TCP/IP protocol suite is widely used for connecting private hosts and networks to the Internet, it has some inherent security problems [20] . Several security vulnerabilities and weaknesses are existing in its specifications and in some of its implementations. The vulnerabilities found in the TCP/IP protocol and its implementations

could enable intruders to attack TCP-based systems. Methods that may be employed by intruders to exploit network's vulnerabilities, include: prediction of TCP sequence number, gaining access to existing TCP connections via spoofing of IP addresses [17, 21] , misuse of IP's source routing principle, causing denial of service to other users using Internet Control Message Protocol (ICMP) messages, etc. It can be seen that security flaws in TCP can be hazardous for computer network, by considering the fact that most of the important application programs such as Simple Mail Transfer Protocol (SMTP), telnet, r-commands (rlogin, rsh, etc), File Transfer Protocol (FTP), etc. are using TCP as their transport layer. Some work has been done to identify and analyze vulnerabilities of TCP/IP and in order to develop security enhancements to overcome these flaws [2] . This involves:

- Analyzing the state-transition diagram of TCP and determining the security significance of some "improperly-defined" transitions between the different states.
- Determination of the importance of timers in different states and associated security problems, if a state does not have the necessary timer-backup or escape route.

The TCP state-transition diagram is analyzed using a "reverse engineering" technique called slicing[2] [22] . Slicing was employed to filter out the relevant state-transition information from the TCP source code, where, a file of 1700 lines of C code implementation of the state-transitions in the TCP including all header file definitions has been reduced to a

[2] Program slicing is defined as an abstraction mechanism, in which code that might influence the value of a given variable at a location is extracted from the full source code of the program.

manageable size of approximately 180 lines sliced code. This sliced code contains only the relevant state-transition information, which aided in abandoning unnecessary information to simplify the code. Using this process, extraneous-state transitions present in some implementations of TCP have been discovered. Several phony state-transitions have been determined in a number of TCP implementations such as SUNOS 4.1.3, SVR4, and ULTRIX 4.3, where the presence of these transitions has not been defined in the TCP protocol specification. Using this approach, various sequences of packets can be identified to be potentially hazardous to the security system of the network. These TCP vulnerabilities can be exploited by intruders [2] .

In this section, we will explore the fundamental weaknesses of the TCP/IP protocol suite, and discuss ways to defend against the TCP/IP attacks and their effectiveness.

2.3.3.1 IP Spoofing

IP Spoofing is defined as the act of creating a forged connection to a computer system. If spoofing of IP addresses leads to unauthorized remote root access to systems behind a router-based firewall, then IP address authentication based applications can be exploited. The two weaknesses that are explained below are found in the Transport and Network Layers. They can be exploited to create a forged connection to a computer system, by someone who understands how the TCP/IP protocol suite establishes a connection between two computer

systems. The information given can be used to initiate an attack known as IP spoofing on network hosts that use the TCP/IP protocol suite [17] .

- In the Transport Layer, to keep track of the next packet expected to be received, the TCP sequence numbers are used. This numbering presents a weakness, because of the regularity of the initial sequence numbers, which allows the prediction of sequence numbers.
- In the Network Layer, authentication is not used with the machine IP addresses, therefore, forged IP addresses are accepted by the TCP/IP protocol suite.

As an example to see how IP spoofing is actually performed, we will assume that there are two hosts, host A and host B, where host A trusts host B. To establish a legitimate connection between host A and the trusted host B, they follow the three-way handshake mechanism of TCP/IP as depicted below [2] :

1. $A \rightarrow B$: SYN (Seq. no. = M)
2. $B \rightarrow A$: SYN (Seq. no. = N), ACK (Ack. no. = M+I)
3. $A \rightarrow B$: ACK (Ack. no. = N+I)

Now, let us assume that there is a third host C, which is controlled by an intruder. If host A is granted some special privileges by host B, then host A can get some actions performed by host B. If host C wants to get for itself the same action done by host B, it has to establish

a forged connection with host B, and prevent host A from informing B of any failure in the authentication system of the network. Host C achieves this by spoofing the IP address of A so that host B will believe that he is actually talking to host A. Therefore the following steps are followed by host C:

1. Host C starts by sending a number of probe packets to B and A, in order to determine whether there exists any kind of trust relationship between hosts A and B. For this purpose, commands such as showmount, rpcinfo, and finger can be utilized.

2. After that, in order for host C to prevent host A from responding to the packets sent by host B, it either floods host A with incomplete connections or simply waits for host A to go down for any reason.

3. Next, host C sends a number of connection-requests to host B and cancels them after receiving the SYN-ACK packets from B. By doing so, host C can determine the behavior of B's TCP sequence-number generator. The periodicity of these numbers is determined, and used by host C to generate and send a forged packet to B with a forged sequence number.

4. Finally, the three-way handshake mechanism for this illegal TCP/IP connection is performed as follows:

   a. C $\rightarrow$ B: SYN (Seq. no. = M), $\Longleftarrow$ Source address = host A

b. B → A: SYN (Seq. no. = N), ACK (Seq. no. = M+1)

c. C → B: ACK (Ack. no. = N+1), ⇐ Source address = host A

2.3.3.2 Extraneous State Transition

Extraneous state-transitions are transitions that exist between TCP states without being defined in the TCP specification. Extraneous state-transitions which may lead to severe security violations of the system, exist in several implementations of TCP.

As an example, consider the following situation, where a host C sends a packet to another host A, with both SYN and FIN flags set. Then, host A responds by sending an ACK packet back to host C, as shown below [2] :

C → A : SYN FIN (Seq. no. = M)

A → C : ACK (Ack. no. = M+I)

Examining the state-transition diagram in Figure 2.7, it can be observed that host A is initially in state LISTEN. It processes the SYN flag of the received packet from host C, and perform a transition to the SYN-RCVD state. Next, it processes the FIN flag and performs a transition to the state CLOSE-WAIT. Because, a transition from SYN-RCVD state to the CLOSE-WAIT state is not defined in the TCP specification, this transition would have been considered as a normal transition if it had been made from the ESTABLISHED state.

In this attack scenario, the TCP connection does not get fully established because the three-way handshake does not complete. Therefore, the corresponding network application never get the connection from the kernel. When, host A's TCP is in CLOSE-WAIT state, it remains in the socket-listen queue waiting for the application to send a close signal, so that it can send a FIN packet to host C and terminate the connection. However, the application will not send any message to help TCP perform any state-transition. Thus, unless the keep-alive timer is enabled, A's TCP remains stuck in the CLOSE-WAIT state. With this timer enabled, TCP can reset the connection and perform a transition to the CLOSED state, but after a period of usually two hours . If an intruder-controlled host C does not send any more packets to host A, it will prevent host A from responding to unexpected SYN-ACKs from other hosts for as long as two hours, thus preventing any TCP state-transitions in host A.

2.3.3.3 Problems with Timers

One problem with timers is related to connection-establishment timers. A connection-establishment timer is turned on, whenever the process of connection setup is in progress. Therefore, if the connection does not get established within a predefined time, TCP returns back to the CLOSED state. For instance, if a connection does not get established within a predefined time, normally 75 seconds, the connection is cancelled. Thus, the server port will not be able to respond for a duration of 75 seconds or as long as the predefined time.

As an example of attacks based on connection-establishment timers, an intruder-controlled host C can stall the login port of host by sending series of SYN packets and not responding with ACK packets to the corresponding SYN-ACK packets from host A to host C. Host C, can then take advantage of this stall time and perform the desired attack, such as IP address spoofing of host A [2] .

Another problem with timers arises during the establishment of a simultaneous open connection between two hosts. This is related to the behavior way of the connection-establishment timer.

As an example, considering two hosts A and B, the first host A will send a SYN packet to start a connection with host B, and wait for a SYN-ACK packet back in response. If almost at the same time, the second host B decides to start a connection with host A, it sends a SYN packet to host A, and wait for a SYN-ACK packet back in response. Therefore, both A and B will send a SYN-ACK packet to each other in response to the reception of the SYN packet from the other party. Next, when each receives the SYN-ACK packet from the other party, it assumes that the connection is established. However, the connection-establishment timer is switched off right after the SYN packet had been received from the other host. These steps are summarized as follows [2] :

$B \rightarrow A$: SYN (Seq. no. = M)

$A \rightarrow B$: SYN(Seq., no. = N)

B → A: SYN(Seq. no. = M), ACK (Ack. no. = N+I)

A → B: SYN(Seq., no. = N), ACK (Ack. no. = M+I)

To see the security relevance of the problem that exists in case of simultaneous open, let us analyze the sequence of steps followed by an intruder-controlled host C and another host A, as indicated below [2] :

1. Host C sends an FTP request to host A to establish a TCP connection between the two hosts to transfer control signals. Next, host A sends a SYN packet to host B to establish a TCP connection for data-transfer and performs a state-transition to SYN-SENT state.
2. When host C receives the SYN packet from A, it responds back with a SYN packet.
3. When host A receives the response packet it assumes that a simultaneous open connection is in progress and:

    a. sends out a SYN-ACK packet to host C,

    b. switches off the connection-establishment timer,

    c. makes a state-transition to state SYN.RCVD.

Since host A is expecting a SYN-ACK from host C, it will get stalled in the state SYN-RCVD as long as host C does not send back the expected SYN-ACK packet. Thus, the

intruder-controlled host C is successfully able to stall a port of host A causing a denial-of-service attack [2] .

2.3.3.4 Protection

It has been noticed that in order for intruders to perform network intrusions, they must follow some sequence of steps specific to the type of TCP attack employed. Such sequences of steps are referred to as attack "signatures", which can be determined by any "network-snifter". Therefore, to maintain the system's integrity, a sniffer program should be installed on the network to report possible intrusions to the system's security administrator who in turn can take the necessary security measures [2] .

2.4 Analysis Tools

Analysis tools enable network administrators to audit their systems. Some tools perform auditing and check for well-known security holes, while others establish databases of checksums of all of the files in a system to allow the system administrator to watch for changes to those files, some tools do both. In the following subsection we will briefly introduce SATAN analysis tool which if not used by network administrators, it would be the hacker's best friend [23] .

2.4.1 SATAN

The computer program called SATAN, is an acronym for Security Administrator Tool for Analyzing Networks. It points out security weaknesses in the network. It tries to crack the network from the outside, like a real hacker would. It is designed to be simple to use and as a security tool for network managers. It helps administrators to find holes in their networks and repair them before hackers can find them, possibly, using the same tool [23, 24] .

In April 1995, the developers of SATAN program made it freely available over the Internet. SATAN does not damage systems that it probes, but simply checks to see if security holes exist and reports back its findings. But it is distributed with its complete source code, which would enable experienced programmers to use it to break into networks. The knowledge of known network security holes is built into SATAN and the product can easily be modified to hunt for new security flaws when they become known [24] .

There is some concern that users will use SATAN to learn about network security in order to better break through system security devices [24] . SATAN makes it much easier to learn how to break into a system. It holds the promise of making systems more secure, provided that a site's administrators run SATAN–and act on its results–before an unauthorized hacker does [24] .

SATAN offers a comprehensive program for network administrators to quickly learn about and repair a variety of security intrusions. The comprehensiveness of SATAN product could be an asset to administrators who rarely have the time to monitor all the security and virus warnings that are periodically dispatched. Using this single application, an administrator could discover various security holes and have an opportunity to repair them. And the fact that the software resembles an intruder provides a better vision of how secure a system is from outside attacks [24] .

2.4.1.1 How does SATAN work?

In order to determine whether or not a host or set of hosts in a subnet are alive, SATAN uses **fping** via a target acquisition program. This target list is then passed to an engine which drives the data collection and the main feedback loop. A list of tests/probes is run against any host not seen before (the set of tests depends on the distance the host is from the initial target and what probe level has been set). Each test generated data record has the hostname, the test run, and any results found from the probe, these records are saved in files for following analysis. HTML is used in the user interface to link the often vast amounts of data to more easy and consistent results that the user can readily digest and understand [25] .

SATAN scans the selected primary computer and optionally all other computers on the same network. It looks for some defined set of vulnerabilities and when one is found, it recommends the proper action to remove it. The following predefined set of vulnerabilities

is quoted from "Tutorials - Security problems" with very minor and limited modifications, where more details and suggestions on what to do to fix them can be found [26] :

1. **FTP vulnerabilities:** In WU-FTPD, where there is a race condition in the code, as well as a bug in the SITE EXEC command, that allows anyone (remote or local) root access on a host running a vulnerable FTPD daemon. Support for anonymous FTP is not required to exploit this vulnerability.

2. **NFS export to unprivileged programs:** An NFS request is nothing but a network message. Any user can run a program that generates arbitrary NFS requests. When an NFS server accepts requests with AUTH_UNIX authentication from unprivileged user programs, a malicious user can execute file access requests on behalf of any user. Because, with AUTH_UNIX authentication, the user identity is nothing but a few user and group ID numbers in a network message.

3. **NFS export via portmapper:** For efficiency reasons, most NFS export restrictions are enforced by the mount daemon. Individual file access operations are handled by the NFS daemon, and the origin of such requests is examined only in special cases such as remote superuser access. Instead of talking directly to the mount daemon, a malicious NFS client can ask the server's portmapper daemon to forward the request to the mount daemon. When the mount daemon receives the request from the portmapper, the mount daemon will believe that the request comes from the file server, and not from the malicious client. When the file server exports file systems to itself (for example,

because the server is a netgroup member) the mount daemon grants access and replies with a file handle. The portmapper forwards the handle to the malicious client. From now on, the client can talk directly to the server's NFS daemon to access the directory and all files below it.

4. **NIS password file access:** Many NIS implementations provide no access control. Every host that asks for information will receive a reply. In order to perform a query, one needs to know the server's NIS domain name. Often, this name is easy to guess, or it can be obtained via the bootparam network service. When the local network is accessible from other networks, a remote intruder can collect password file information and run a password guessing program. Many people tend to choose passwords that are easy to guess.

5. **REXD access:** A request for remote command execution contains, among others, the command to be executed, and a user and group id. By default, the rexd server believes everything that the client sends it. An intruder can exploit the service to execute commands as any user (except perhaps root). The typical rexd server has no protection against abuse: most implementations have no provision for access control, nor do they require that the client uses a privileged network port.

6. **SATAN Password Disclosure:** It is important that the current SATAN password is kept secret. When the password leaks out, unauthorized users can send commands to the SATAN HTML server where the commands will be executed with the privileges of the

SATAN process. SATAN generates a new password every time it is started up under an HTML client, so if there is a suspicion, the program can simply be restarted. SATAN never sends its current password over the network. However, the password, or parts of it, may be disclosed due to flaws in HTML clients or due to weak protection of the environment that SATAN is running in. One possible scenario for disclosure is:

When the user selects other HTML servers from within a SATAN session, some HTML client programs (Netscape and Lynx) disclose the current SATAN URL, including SATAN password information. The intention of this feature is to help service providers find out the structure of the world-wide web. However, the feature can also reveal confidential information. With version 1.1 and later, SATAN displays a warning when the HTML client program exhibits this questionable feature.

7. **Sendmail vulnerabilities:** With almost every sendmail version that was built before February 1995, a malicious user can gain unauthorized privileges by exploiting newlines in command-line arguments or in the process environment. Intruders need to have access to an account on your system to exploit this problem. In addition, pre-8.6.10 versions of sendmail that support IDENT (RFC 1413) functionality have a problem that could allow an intruder to gain unauthorized access to the system remotely (that is, without having access to an account on the system).

8. **TFTP file access:** When the TFTP daemon does not limit access to specific files or hosts, a remote intruder can use the service to obtain copies of the password file or of

other system or user files, or to remotely overwrite files.

9. **Remote shell access:** When the remote login/remote shell service trusts every host on the network, a malicious superuser on an arbitrary host can gain access as any user (except perhaps root). Once inside, the intruder can replace system programs or configuration files (such as the password file) and take over the machine. In addition, there are guest or administrative accounts that might not have passwords protecting the account, which allows anyone to remotely login as that user and gain access to the host.

10. **Unrestricted NFS export:** When a file system is exported without restriction, an intruder can remotely compromise user or system files, and then take over the machine. For example, an intruder can remotely replace a system program or configuration file. For instance, in UNIX, an intruder can remotely install a ".rhosts" file to obtain interactive access. An intruder can remotely install a ".forward" file to obtain non-interactive access.

11. **Unrestricted X server access:** When the X server permits access from arbitrary hosts on the network, a remote intruder can connect to the X server and:

    a. Read the user's keyboard, including any passwords that the user types,

    b. Read everything that is sent to the screen,

    c. Write arbitrary information to the screen,

    d. Start or terminate arbitrary applications,

e. Take control of the user's session.

12. **Unrestricted Modem on the Internet:** Anyone can use the modem to dial anywhere, enabling them to attack random targets and incurring the owner a potentially large phone bill.

13. **Writable FTP home directory:** When the FTP home directory of a UNIX host is writable, a remote intruder can upload a ".rhosts" or ".forward" file to gain access to the system, or may be able to replace files. When a PC (DOS or MAC) permits anonymous users write access to its file system, a remote intruder may be able to replace arbitrary programs or configuration files, or corrupt the file system by filling it up.

2.4.1.2 How to Protect Against SATAN

Detecting that some one is running SATAN against a given computer or network is not easy. One program called Courtney was written to detect SATAN, but it is far from foolproof. While it is very difficult to detect the lighter SATAN scans, the heavier ones can be best detected by running Wietse's tcpd wrappers and examining the logs. Also, some of the SATAN probes output messages to the console. Therefore, if users notice odd messages on their console screen, they should take them seriously [27] .

2.5 Protection Methods

Security protection methods are basically concerned with ensuring network's efficiency and effectiveness. With successful security implementations, risks can be reduced but not eliminated. There are several protection methods to ensure confidentiality, integrity and continuity. The dominating security protection method in the mainframe computing environment is the Access Control. It consists primarily of functions related to:

1. Access Mediation via connection control establishment,
2. Identification by means of Logon-Ids,
3. Authentication by means of Passwords,
4. Different levels of authorization controlled by Access Privileges,
5. Monitoring and enforcement,
6. Disaster recovery programs to respond to incidents,
7. Logging to record traffic and usage of services.

In an open-networked environment such as the Internet, more work should be done and integrated in order to achieve successful security controls. Integrated security should include Physical security, Computer based security, Encryption and Authentication, Procedures, and Awareness [6, 28] :

2.5.1 Password Authentication

The most common security technique within computer systems is the validation or authentication of passwords which is used to validate users. The user should be the only one to know his password, while guessing it randomly should be costly. Additional passwords may be required by the different Internet services. This protection technique will only work as a useful identification method, if passwords are protected and applied correctly. For better protection of a personal password, the following may be used:

1. The minimum length of a password should not be less than six (eight is better) alphanumeric characters,
2. Care and creativeness should be used when constructing a password,
3. Passwords should not be exposed or shared with any one,
4. Passwords should be changed frequently and regularly.

2.5.2 User Authentication Techniques

Normally, user authentication is the first, if not the only line of defense for many computer systems. In most systems, user authentication is based on a user name and a password, where the same static password is usually used every time a user logs in to the computer. Therefore, protection of computer systems against unauthorized uses depends

entirely on keeping the password secret. However static passwords are vulnerable to interception and guessing attacks.

Alternatively password systems can be based on one-time passwords, where a new password is required for each authentication and cannot be used again. With one-time passwords, interception attacks can be eliminated. In addition, cryptographic functions can be used to prevent guessing attacks by producing pseudo-random results of the one-time passwords to make them more difficult to anticipate. One way of implementing a one-time password authentication can be obtained with a device called a smartcard which provides users with a new password for each time authentication. A commonly used commercial smartcard system that provides users with one-time passwords displayed on a credit card sized device known as a SecurID card.

The SecurID system is based on the principle that authentication should be based on something you know and something you hold. The authentication process requires the user to provide something that he knows, which is a four-digit Personal Identification Number (PIN) and the current six-digit Pseudo Random Number (PRN) displayed on the smartcard card which the user holds. The PRN is a continuously changing number generated by the SecurID Algorithm. The PIN /PRN combination is known as a passcode. After the user has authenticated himself, the same algorithm is run by the SecurID database system to check if PIN /PRN combination that is provided by the user is valid for that time of the day. If the combination is determined to be valid and was not used before, then the user is granted

access. Disallowing a passcode to be used more than one time prevents eavesdropper from using an intercepted passcode in a replay attack.

The SecurID system is the first line of defense for some computer systems where users are first required to authenticate themselves to the SecurID system and then go through the standard user authentication scheme on the host they want to connect to.

2.5.3 Encryption

Encryption is one of the protection mechanism for data confidentiality and integrity. It also, via secret handshaking, can ensure the correctness of communication between any two computers. It makes key resources unavailable to attackers, therefore, when properly implemented, it can prevent attacks against networks and protect against information disclosure. Encryption is becoming the standard policy in any communication of sensitive information. It has many applications including: the protection of file servers, hard disks, faxes, telephone conversations via other types of communication. But, it is particularly important for communications with remote sites. Many applications have their own encryption schemes which may be transparent to users, as long as they operate with the same applications at both ends of the transmission within restricted environments.

Digital Signatures can also be created using encryption algorithms in order to verify the identities of the senders of the messages as well as their recipients. Public keys can also be obtained through the Internet.

On the Internet, a message will be stored on several, may be thousands, of servers before it reaches its final destination. Therefore, messages on any one of those servers can be looked at or modified but it is difficult to do so because a message is broken into parts. This practice can be made almost impossible with encryption.

2.5.4 Protection with Firewalls

The best line of defense is an up-to-date and constantly maintained firewall. A firewall/proxy server is a mechanism that is used to protect a trusted network, such as an organization's internal network, from an untrusted network, typically the Internet, or any other untrusted network [3] . Firewall/Proxy servers provide the most reliable method to control outbound access and to protect networks against unauthorized intrusions. It checks addresses and characteristics of messages to make sure that they follow authorization rules. All messages that are verified to be legitimate are allowed to flow through the firewall, while others are blocked. The majority of firewalls are used between internal networks and the Internet, but they can be used in any internet, such as a company's wide area network [3] . The design decision sets the general attitude of the firewall whether to provide a higher degree of service or a higher degree of security. To protect the firewall server itself, no users should be allowed to login on the firewall server [29] .

2.6 Security and Policies

Security is needed to prevent destruction of data by an intruder, maintain the privacy of local information, and prevent unauthorized use of computing resources. There are three levels of security policies: (1) organizational policy, (2) network service-access policy, and (3) firewall design policy [30] .

1. Organizational level policy is the highest policy level. It deals with general overall guidelines and directives that should be followed and satisfied by lower level policies.
2. On the next level of security policies, the network service access policy is formulated. It is considered as a higher-level, issue specific policy that define what services will be allowed or explicitly denied from the protected network, how they are used, and exceptions to this policy. The following lower-level policies are addressed:

    a. Site-specific policies concerning physical access to properties,

    b. The general access policy to information systems,

    c. Policies related to the specific access of services on the information systems.

3. Firewall design policy is a lower-level policy that describes the actual ways for restricting the access and filtering the services by the firewall as defined in the network service access policy. The firewall security policy should clearly reflect the importance of strong firewall administration. Therefore, if the firewall is not

administered appropriately, it is most likely to become insecure, and allow break-ins, while at the same time, it is believed that the site is still secure.

Designing, installing, and using a firewall system to achieve network security is directly influenced by two levels of network security policies: network service access policy and firewall design policy [3] .

2.6.1 Organizational Policy

At the highest level, the overall organizational policy might look like the following [3, 31] :

- Information is critical to the economical growth of the organization.
- In order to ensure the confidentiality, integrity, authenticity, availability and utility of information, all cost-effective efforts shall be made.
- It is the priority for all employees at all levels of the company to protect the confidentiality, integrity, and availability of information resources

The design of network and firewall security policies is greatly affected by the type of security environment for which it is designed. In this discussion, we will focus on the corporate and academic security environments. Corporate and academic institutions face different concerns related to the security of their information and computing resources.

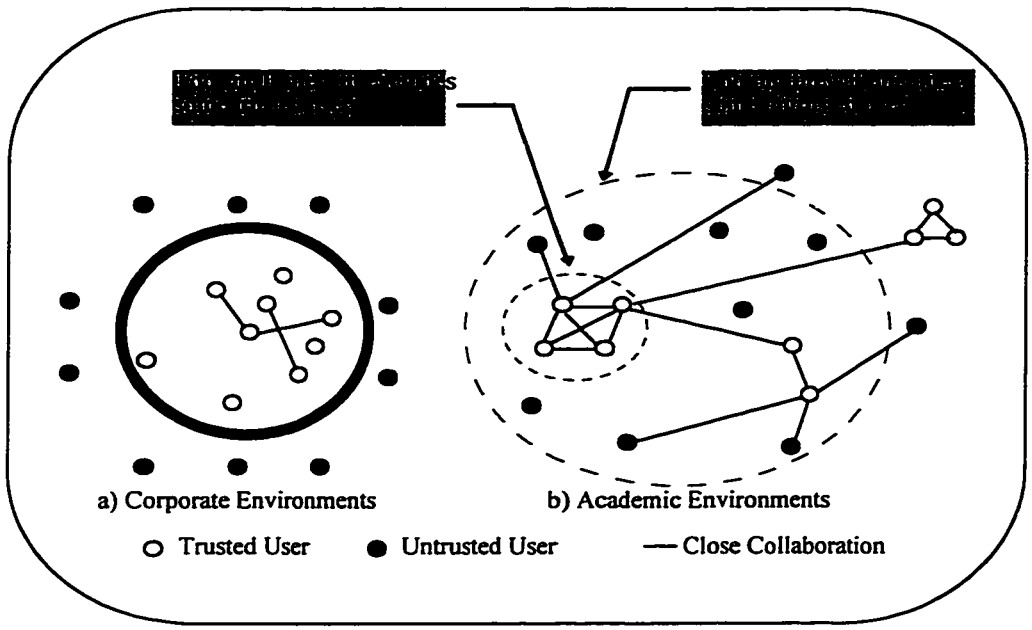


Figure 2.8. Nature of Trusted Users, Untrusted Users, and Collaboration in Corporate and Academic Environments.

In most corporate security environments, firewalls are used to block or heavily restrict access from untrusted hosts to internal data and computing resources as well as to limit access from the inside to untrusted hosts outside the corporate. the following is typical in the corporate security environment [31] :

- A corporate firewall is defined as a strong security perimeter around collaborating employees within the corporation, as shown in Figure 2.8(a).
- The network security perimeter surrounds the corporate network and exclude everything else, except in some cases, where it might include machines at employee homes.
- The security perimeter cautiously controls the transfer of information and in sometimes

it actually forbids all outward information flow.

- Although, more open access to the Internet might be desired by many corporations, often, they choose to limit Internet access to achieve corporate security by sacrificing services such as personal World-Wide-Web pages.

- In recent firewall designs some of these limitations are relaxed, but only to support specific interactions between sites that are located within a single private network or authentication domain.

On the other hand, for academic research groups, the trade-off between safety and collaboration is unacceptable. Consequently, the traditional corporate firewall is not suitable for academic environments, as pointed out below [31] :

- The natural place to draw a security perimeter in a corporate environment is around the whole corporation. However. this can not be the case with academic environments, as shown in Figure 2.8(b), where it seems nearly impossible to draw a perimeter surrounding everything a trusted user might need to interact closely-with, while keeping other untrusted users out.

- If a corporate firewall is placed around a research group, it would exclude collaborators located in other sites. Therefore, If the firewall is chosen to be:

  * too big. it will definitely include untrusted people, especially that universities offer

almost no physical security. This is indicated by the dashed box of Figure 2.8.

* too small, then it will exclude some of the people with whom we need to share data as indicated by the dotted box of Figure 2.8.

Corporations can tolerate limitations of the Internet connectivity in order to keep their sites secure, however, research organizations cannot perform under such limitations, for the following reasons [31] :

1. Researchers and trusted users need unrestricted access to Internet resources located outside the firewall.
2. Researchers and trusted users need an unrestricted ability to publish and distribute information to the world outside the firewall, which is critical to the research community.
3. Research collaborators and trusted users located outside the firewall should be allowed to access protected resources.

In addition, there is a number of other factors that may apply when comparing the two security environments, as shown in Table 2.2. These factors include the usual considerations of cost, ease-of-management, performance, and reliability in a heterogeneous computing environment. Usually, academic institutions can not allocate enough money to obtain the right computer security hardware, software, or personnel. Therefore, leading to the lack of dedicated security staff which in turn could mean for instance, that a firewall might be managed by inexperienced people such as new graduate students with limited expertise.

Comparison Factors	Corporate	Academic
Firewall security perimeter	A corporate firewall is defined as a strong security perimeter around collaborating employees within the corporation that excludes everything else, except in some cases, where it might include machines at employee homes.	An academic firewall seems nearly impossible to draw a security perimeter surrounding everything a trusted user might need to interact closely-with, while keeping other untrusted users out.
Security perimeter controls	Cautiously controls the transfer of information and in sometimes it actually forbids all outward information flow.	Supports free exchange of information.
Limitations of Internet access	Corporations can tolerate limitations of the Internet connectivity in order to keep their sites secure. These limitations can be relaxed, but only to support specific interactions between sites that are located within a single private network or authentication domain.	Research organizations require unlimited Internet connectivity in order to support collaboration between internal and external research groups.
Financial allocations	Usually, adequate financial allocations can be made available to obtain the right computer security hardware, software, or personnel.	Usually, academic institutions can not allocate enough money to obtain the right computer security hardware, software, or personnel.
Expertise	Dedicated security staff.	Lack of dedicated security staff which results in relying on inexperienced people such as new graduate students with limited expertise.

Table 2.2. Comparison of Corporate and Academic Security Environments

2.6.1.1 Centralization Versus Distribution

Usually, in an enterprise networking environments, users and resources tend to reside in more distributed locations. Often, it is desirable to centralize operations to some extent to reduce the overhead but more importantly, security risks that could be caused by the reliance on management by multiple inexperienced system administrators. But, this does not necessarily mean that security information has to be stored centrally, for example, it is often advantageous to keep access control lists (ACLs) with their related resources, while still be managed centrally. Doing so, it will be possible to improve the performance of access control checks as well as allowing the ACLs to be moved with the related resources from one system to another [32] .

A large enterprise network may be subdivided into a number of management domains, in which authorized users in a given source domain may utilize resources in another target domain. Therefore, it is necessary to set up inter-domain relationships, so that both domains can be used to obtain the required information for user authentication.

2.6.2 Network Service Access Policy

Network service-access policy should be an integral part of a strong site-security policy and an overall policy[3] for the protection of information resources. In network service access policy, the attention is directed toward the restriction and use of internetwork services. But

[3] This should include everything such as document shredding, virus scanning, remote access, and floppy disk tracking.

while doing so, it is very important that the network service access policy also includes all other means of network access such as dial-in and serial line Internet protocol / point to point protocol (SLIP/PPP) connections. When restrictions is implemented on one network service access, users may be motivated to try other ways. For example, if web browsing is prevented through a restricted gateway access to the Internet, users may try to obtain this service by creating dial-up PPP connections which are, normally, authorized as an ad hoc connections. These ad hoc connections are likely to have an unacceptable security effects resulting in opening the network to attacks.

The network service-access policy should be realistic and reliable, as well as designed before the actual implementation of the firewall. As realistic policy, it should maintain a balance between protecting the network from known risks and providing users with reasonable access to network resources.

Adherence to the network service-access policy prevents ad hoc circumvention or modification of the firewall's access controls. The typical network service-access policies that a firewall should implement are [3] :

- Allowing access from the internal site to the Internet, and disallowing access to the site from the Internet
- Allowing limited access from the Internet to selected systems such as e-mail and information servers.

Normally, firewalls implement the policies of network service-access that allow some access from the Internet to selected internal servers. But this access is accepted only when necessary and combined with strong authentication.

2.6.2.1 Integration

In an enterprise network, many of the interconnected heterogeneous computer systems may have their own local user authentication and access control facilities. The usability and manageability of enterprise networks are greatly improved, when integrating local system security facilities for the whole network. Some of the problems that may arise in network environments that have distributed security facilities include the following [32] :

- Because each system recognizes its own set of local users and assigns them unique IDs, passwords, and other security-relevant attributes, a user has to remember multiple ID-password pairs. Therefore, users may start writing passwords down, and subsequently, these written passwords may fall in the hands of some who might endanger the security of the system.
- Because resources access is governed by local access control facilities which depends on local user IDs and security attributes, users must have local IDs defined on each system, so that they can access resources on different systems.

2.6.3 Firewall Design Policy

The firewall design policy defines the rules used in implementing the policy of the network service access. The designer of firewall design policy must be fully aware of the threats and vulnerabilities related to TCP/IP, and the capabilities and limitations of firewalls in general. It is extremely important to consider a security design policy before implementing the firewall. Normally, firewalls implement one of two basic security design policies: (1) permit all services unless otherwise are expressly denied, (2) deny all services unless otherwise are expressly permitted. Services which should not be passed through the firewall can be placed, separate from other site systems, on screened subnets. Depending on security requirements, some firewall types are more appropriate than others. A firewall that implements the first policy would have the following features [3] :

1. By default, allows all services to pass into the site, except services that are identified as prohibited by the service-access policy.
2. Offer more ways for getting around the firewall which is not desirable most of the time. An example of this is that users would be able to access new services that are not addressed or not yet denied by the policy. Also, denied services could be run by users at non-standard TCP or UDP (User Datagram Protocol) ports that are not specifically denied by the policy.
3. Better accommodate services that are difficult to filter, such as X Windows, FTP, Archie,

and RPC (Remote Procedure Call) [33, 34] .

On the other hand, the second policy is stronger, safer, and follows the standard access model that is used in all areas of information security. Therefore a firewall that implements this policy has the following features [3] :

1. Deny all services by default except those that have been identified as allowed,
2. It is more difficult to implement,
3. It is more restrictive for users,
4. Services such as X Windows, FTP, Archie, and RPC may have to be blocked or heavily reduced.

In order to design a firewall policy and implement that policy, the policy designer should start with the most secure firewall design policy, which denies all services except those that are explicitly permitted. Then he should understand and try to answer the following questions [3] :

1. What Internet services will the organization be using and what risks are associated with providing these services and access?
2. Are there any additional requirements, such as encryption or dial-in support?
3. Will the services be used on a local basis, across the Internet, dial-in from home, or from

remote organizations?

4. What is the cost of providing protection, in terms of controls and network usability?

5. If a particular service is too risky or too expensive to secure, how would the decision be made regarding security versus usability?

In this section, we will discuss a selection of a number of design considerations that should be considered when designing security solutions for enterprise networks.

2.6.3.1 Layering

Security mechanisms can be implemented at different layers in the protocol stack allowing different levels of flexibility and transparency. Generally speaking, lower layers implementations are more transparent to applications. On the other hand, implementations at higher layers provide more flexibility of customization to the requirements of individual applications. Applications often implement security controls into their own specific protocols, for the following two reasons [32] :

1. Existing protocols have a very limited security,

2. In several cases, the decision is made to build security into an application in order to make it portable to environments running different sets of underlying protocols.

This approach may lead to multiple security mechanisms that may provide the same service without interacting with each other. Therefore, as much as practical, secure application-layer communication protocols should be used to support a wide range of applications.

2.7 Summary

In this chapter, we defined the important terms that are used in this thesis. We have, also, introduced the Internet and its security related aspects. TCP/IP, the protocol that is used in Internet communication, was presented and its security problems were discussed. Also, we have discussed security analysis tools where we provided more detailed explanation of SATAN security analysis tools. In addition, we discussed the various protection methods. And finally, we discussed the different security policies.

CHAPTER 3

FIREWALLS

A firewall is a trusted system that is placed between a trusted internal network and another untrusted external network. The firewall system implements a policy that defines what information should be allowed to pass through. In general firewalls have the following features and limitations [6] :

- **Features:** Firewalls can be used to enforces the security policy, control the access to the protected network and provide one central point of security. They can provide more privacy by hiding addresses. Firewalls, also, provide logging for security and other purposes. They can, also, notify the network administrator of security related events, so that he can take the appropriate actions. In addition, they can be integrated with authentication keys.

- **Limitations:** Firewall limitations include: restricted access to desirable services, back door access problem, inside attacks, e-mail viruses, potential bottleneck, and single point of failure.

3.1 Overview

A firewall should have a fire door that may allow certain types of traffic to flow in one direction only (outward from the protected network). In this case, it must prevent all traffic from entering the network at all costs, and it must restrict outbound traffic to an acceptable level. On the other hand, A fire door that facilitates two-way passage may allow the fire to spread. Many firewall products on the market introduce this danger by allowing selected traffic into the network. When we allow administrative controls to permit passage in both directions, we are potentially opening the fire door [4] .

Although the configuration may be perfectly acceptable for some applications, a firewall that allows two-way access is probably not effective. The security of the firewall is determined by the effectiveness of the controlled access, not the firewall itself [4] .

Firewall system monitors can be implemented to detect potential problems before they become major. Most firewall software includes logging capabilities to record connection attempts, source and destination IDs, and similar data. For instance, one type will log use of application gateways to report who is using the system and what they are using it for. It is important that network managers continuously check the logs for attempted break-ins. Along with providing a level of security, this also assists network managers with capacity planning.

There is a little consistency or standardization in the form and functions of Internet firewalls. Firewalls can be built in several ways, using a variety of mechanisms. Most firewalls perform as a "proxy server" and/or an "applications gateway." Some vendors consider them the same thing since there is no universal agreement on how these roles differ. A **proxy server** usually implies that the firewall performs various network and administrative tasks on behalf of end users on the protected internal network. These tasks may include authentication of the user to a host node on the Internet or translation of addresses so that the internal network remains hidden from the Internet side. As an **application gateway**, the firewall may flip-flop between client and server roles to control certain high-level protocol connections and operations. Some firewalls may perform checking such as network-layer filtering, session-layer filtering, address translation and the ability to act as a domain name server, and checking of passing files or messages to detect viruses or other threatening executables that might infect incoming data [35]. Also, firewalls could be used to separate administrative domains within the private network [32].

Security and ease of use are antagonistic objectives. In other words, the more secure a firewall is, the more it tends to inhibit users from establishing legitimate Internet connections and performing even authorized communications. In fact, until recently, to establish connections, many firewalls require users to follow a two-step procedure. First, users would have to authenticate themselves to the firewall. They would then need to re-authenticate themselves to connect to the desired destination host (server node). To consolidate this

procedure, a new protocol was developed, called SOCKS. The protocol involves a type of proxy operation, where the firewall handles more of the network and administrative processing on behalf of the internal end users. Users can establish connections through a SOCKS firewall more or less transparently, with only a single end-to-end authentication. Some firewall vendors have advanced their systems so that they do not require users to perform an intermediate authentication anymore [35] .

Firewalls, a combination of hardware and software, are designed to permit the flow of desired information while protecting designated resources. They create narrow channels for tracking and controlling information flow. Most firewalls follow a single and simple philosophy: everything not explicitly permitted is denied. Firewalls should be capable of three things: (1) monitor all traffic to and from the protected network, (2) permit only authorized traffic to pass through, and (3) be immune to unauthorized manipulation [33] . Some Internet firewalls are stronger than others, but, just like physical firewalls, computer networks firewalls are not perfect. They also vary in their ability to handle traffic loads. The fastest and slowest of the group vary across a wide range. Sometimes a firewall will slow a response time to the extent that end-users will look for ways around it [28] .

The three types of firewall architectures are router-based packet filters, circuit gateways, and application-level gateways. Packet filters work at the TCP/IP levels. They accept individual packets of information that come from authorized locations or addressed to certain destinations on the network. Because they are fairly low-cost solutions, packet

filters that can be installed as part of an organization's Internet router are a viable option for many organizations. On the other hand, the packet filter approach does not provide a significant level of security on its own [36] .

The idea behind the circuit-level gateway approach to network security is to prevent any direct physical contact between machines on the internal network and the outside world. In this approach, a proxy or substitute address is used for the contact with the outside source. When the information is transferred to the substitute address, the proxy transfers information to the appropriate internal destination. This should prevent intruders by limiting the amount of information individual machines share with the outside world [36] . Circuit-level gateways do not examine individual packets of information, rather they accept multiple packets once the authentication at the connection setup time takes place [36] .

The application-level gateway approach to network security is the most robust of all present methods. It is very similar to the circuit-level model, but, it offers one key benefit since it examines each message as it passes through the gateway. In this way intruders are prevented from breaking security by hiding potentially destructive data among apparently safe data [36] .

3.1.1 Internal Firewalls

The attention has been focused on Internet firewalls, but modern business practices emphasize the importance of internal firewalls. Considering an organization with separate

but connected networks, it may be desirable for some users to have access to more than one network, while on the other hand, it may be unnecessary as well as undesirable for all users to have access to all networks. In a wide area network, application level security may be used to protect sensitive data, however, using firewalls to separate internal networks, greatly reduces security risks [3] .

Internal firewalls can remarkably reduce the threats that are caused by internal hacking, which is defined as the unauthorized access by authorized users. Internal hacking is a problem that continuously exceeds the number of external hacking in all security surveys. Internal firewalls can have an important role to play in enforcing the access control policy when someone outside the organization needs access to some, but not all, of the internal information networks, or when multiple networks designed by different people with different rules are asked to trust each other.

3.1.2 Gateways and the Demilitarized Zones

The term gateway is an important terminology that is often associated with firewalls. Internet firewalls are referred to as secure Internet gateways. However, the term gateway has a more specific use as is shown in Figure 3.1. While, a firewall consists of several different components, such as filters or screens to block transmission of certain traffic, a gateway is a machine or set of machines to provide relay services in order to make up for the effects of the filter [3] .

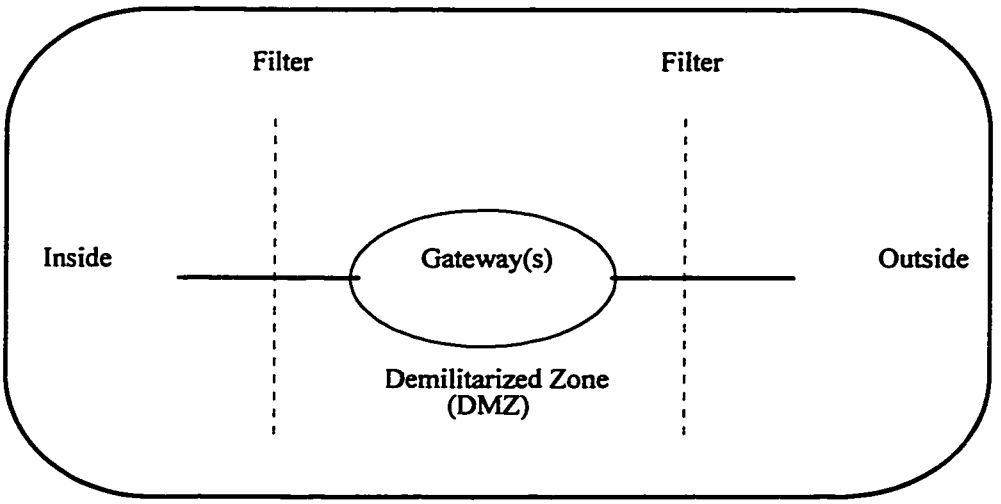


Figure 3.1. Schematic of a firewall

The sub-network having the gateway is often called the demilitarized zone (DMZ). The DMZ-gateway is protected against internal and external attacks by the inside and outside filters respectively. Occasionally, a gateway in the DMZ is used in conjunction with another gateway in the internal network to provide protection in case of a compromised DMZ-gateway [33].

3.1.3 Proxies

Network protocol proxy is a program acting as a gateway that runs on a firewall host to connect specific service requests across the firewall. Proxies exist for a wide variety of services, such as X, FTP, TELNET, etc. A minimal TELNET service proxy is represented by Figure 3.2, in which user's keystrokes are forwarded by the proxy to a remote system, while maintaining audit records of connections. The software on both sides work under the

illusion given by proxies of a direct point-to-point connection. Additional access control and audit may be performed by proxies as desired, since many of them interpret the protocol that they manage. For instance, the FTP export of files can be blocked by the FTP proxy while they can be imported. This represents a granularity of control that is not achieved by router-based firewalls [5] .

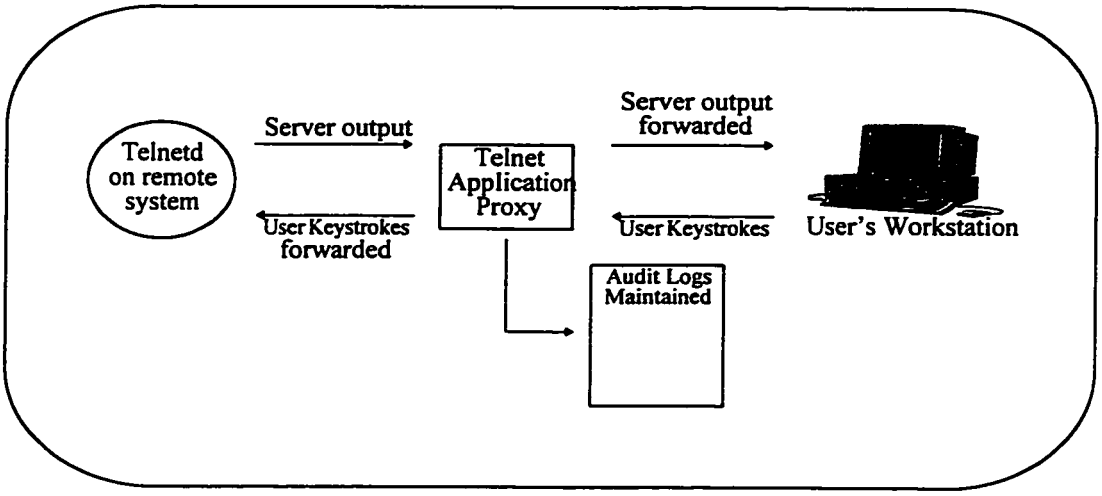


Figure 3.2. An Application Proxy

The most significant security benefit of using proxies is that they provide a suitable point to require authentication. Restricting the incoming traffic to authenticated users only is a good step in the right direction to minimize the effect of security holes that hosts on the private network may have. For example, to connect from the Internet to a protected network, normally, connection to the proxy and authentication to it is made first, and then connection to a host within the protected network is completed. The proxy protects private networks by permitting only authenticated users to gain access from external hosts. It also protects

the firewall host itself as well, by eliminating the need for users' logging into the firewall machine.

3.1.3.1 Other services

There is always a possibility that the kernel may have trapdoors or hidden network services built into it, therefore, every effort should be made to remove unnecessary kernel services at system build time. Other services, such as Internet Simple Mail Transfer Protocol (SMTP) and USENET news, fit in with the proxy approach to firewalls, since they already act as store-and-forwarders. Sometimes, these service daemons run with system privileges, which gives attackers the power to exploit any bug that they may contain. In some situations, the server itself can compromise the security of the network, as in the case of one version of the file transfer protocol daemon "WUArchive" ftpd [37] , which contained a bug that permitted anyone on the Internet to gain super-user access to systems on which it was running. Relying on approximate assessment of privileged systems software for their trustworthiness can be sufficient when using "well known working versions" of common programs such as the FTP server "ftpd", but this is not always the case. This problem can be avoided by using proxies that can operate according to the following rules [5] :

1. Run locked into a specific sub-directory by means of the UNIX command "chroot"[4].
2. Run without special system privileges, to further reduce the chance that they might be

[4] Chroot is a UNIX system call that permanently restricts the root of the working file-system of a process to a given directory structure.

able to damage the system.

Ideally speaking, an external user should not be able to ever interact with a privileged process. In practice, however, the Internet service master daemon "inetd" needs to run with privileges, but outside users cannot interact directly with it.

3.1.4 Tunnels

Firewalls offer strong protection, however, they can be bypassed by tunnels. It is possible to encapsulate a protocol within itself. What of concern to us is that Internet Protocol (IP)[5] may be buried within another IP or some part of its own suite, such as TCP (Transmission Control Protocol) or UDP (User Data Protocol) [16] . If a firewall permits user packets to be sent directly without intervention, a tunnel can be used to bypass the firewall. If two users on both sides of the firewall wish to bypass it, then, the two will construct a tunnel between an inside host and an outside host to allow the free flow of packets [16] . Almost any mechanism such as DNS (Domain Name System) messages or pairs of FTP, can be used to build a tunnel. The extent of the damage done by a tunnel depends on how routing information is propagated. Suppression of routing information is almost as effective as isolation. If an external router knows a path to any internal network other than gateway's, and an application or circuit level gateway is being used, then something is leaking. Often,

[5] Each IP packet contains a 32-bit source IP address and a 32-bit destination IP address in fixed-location fields. Routers routinely examine these to determine where to send the packets they receive[15]

such a situation can be detected. This suggests that a gateway network should not be a subnet of an internal network. Instead, it should have its own, separate class address [16] .

3.1.5 Other Aspects

The aspect of restricting access is not the only one that matters to computer network security. Other aspects include the situation where firewalls may become annoying by requiring end-users to learn special functions in order to walk through the firewall, in which case, end-users may start looking for ways to go around them. Another aspect is the inclusion of "secured" kernels. A secured kernel is defined as a modified version of an operating system so that it contains only the necessary services that are needed to run the firewall. This approach forbids attackers from exploiting other services, simply because they don't exist. The majority of firewall products come with secured kernels based on UNIX variants, while some other product's secured kernels are based on DOS.

There has been trade-offs between UNIX and DOS based products. The lack of integrated networking capabilities in DOS can be seen as an advantage, because it means that the firewall can't be crossed over when the firewall application is compromised. However, DOS based products have shown a slower performance, which is considered to be one of the very important security issues for the following reasons [28] :

- Frustrated end-users may look for ways around the bottleneck. That is caused by the slow response time of the firewall.

- Any path that circumvents the firewall can present a point of entry to intruders.
- When invaders overwhelm a firewall with requests more than it can handle, access to protected resources may be cut off. This is called denial-of-service attack, which is one of the most common forms of attacks on the Internet. The weaker a firewall's performance, the more sensitive it is to a denial-of-service attack.

3.2 Firewall Types

A firewall is simply defined as a collection of components placed between two networks to protect a private network from unauthorized intrusion [12] . Strictly defined, a network firewall does not let traffic pass in either direction, at any time, for any reason. A product based on these principles, however, would be impractical in most cases [4] . As shown in Figure 3.1, a firewall is a filtering mechanism placed between the private network and the outside world so that all incoming and outgoing traffic must pass through it to prevent unwanted and potentially damaging intrusion.

Internet firewalls are relatively new in the computing world, but have their roots in control mechanisms and security measures that have long been a standard practice in the mainframe community. To provide reasonable protection of networks from unwanted attacks, a firewall system should be installed.

If we accept the theory that most data loss can be attributed to user error, we can use internetwork relays to segment the network into domains. Users should not be able to pass

through these relays without proving that they have a specific need for the requested data, and without specific authorization [4] .

Internetwork relays also have the benefit of controlling network traffic. Usually, such devices are little more than robustly configured bridges or routers, contained within the boundaries of the organization's enterprise. In many respects, these linking devices can be thought of as firewalls. They also provide a model for the next level of protection: router-based firewalls. These structures may or may not be true firewalls, depending on how they are configured, keeping in mind that firewalls were designed to allow traffic to flow to desired destinations and block traffic based on specific configuration criteria. Routers used in combination with specially configured hosts, can provide a reasonable degree of protection for some types of networks, at certain levels [4] .

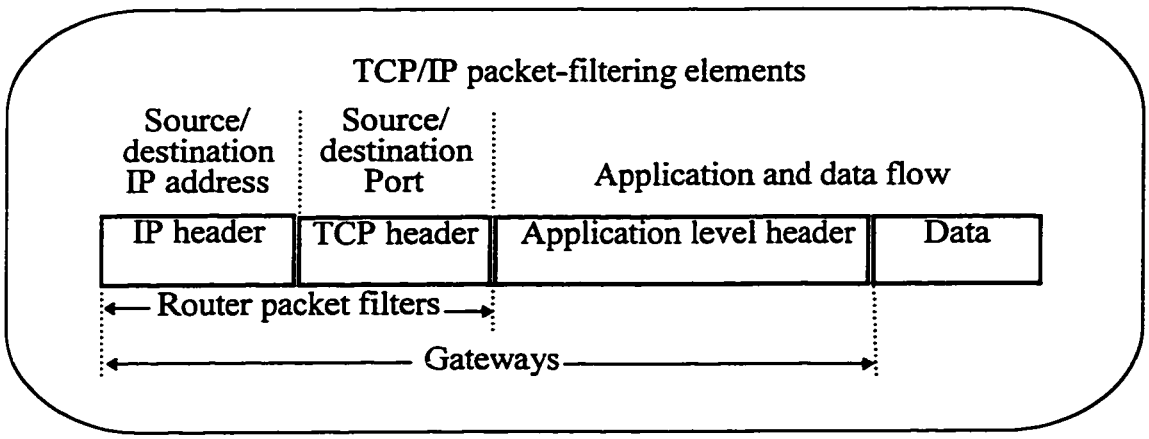


Figure 3.3. TCP/IP packet-filtering elements

Figure 3.3 shows the TCP/IP packet-filtering elements used by router-based and gateway firewalls. Four pieces of information contained in IP and TCP or UDP datagrams, differentiate them from any other connection on the network. These elements are sender's and receiver's IP address of the network interface, and sender's and receiver's port number. The sending port identifies a unique outbound communication channel assigned to a particular client application, while, the receiving port number identifies which server application the packet is to be delivered to for processing. Server applications usually listen to a particular assigned port number, such as port 23 for Telnet. A common method of filtering connections, base the decision on server port numbers to permit connections for "safe" applications like Simple Mail Transfer Protocol (SMTP) and deny connections of dangerous services like FTP or unknown (unassigned) applications. Other sophisticated firewalls can filter traffic based on TCP/IP options headers [38] .

One advantage of isolation networks is that they can also simplify the establishment and enforcement of new Internet addresses, especially for large private networks that may otherwise face the possibility of having to undergo significant reconfiguration.

3.2.1 Router-Based Filters

Router-based firewalls are routing devices on Internet links configured to route data out of a network and to select certain types of data to be routed into the network. Routers[6] alone

[6] A router is a device that is normally used to create a permanent, Internet connection to the outside world (often via a commercial Internet provider)

do not make good firewalls because they were not designed to provide the level of control that the other types of firewall products can. A router examines only one packet at a time and forwards it. The security philosophy used by routers is that: "whatever is not explicitly forbidden is permitted." By allowing two-way communication between a hostile network and a secure one, routers enable intruders to bypass them using forged IP addresses.

The simplest approach to creating a firewall involves using a programmable router. Routers work by controlling traffic at the IP level, selectively passing or blocking data packets based on source/destination address or port information in the packet's header. Routers and mini-firewalls can provide considerable help in minimizing the vast majority of data loss, which occurs as a result of user error. Using routers and bridges in combination with server configuration can ensure that only users who have a legitimate need to access a piece of information can do so [4] .

Router-based filtering is a network-layer security mechanism. It can, for example, prevent any data packet with an unknown source IP address from entering a private network over the Internet connection. Such a broad filter may constrain users on the private network by allowing them to communicate only with those Internet nodes for which the addresses have been explicitly authorized in the router's filter tables [15] .

At the very least, a router can be used as a packet filter. This approach is the most common internetwork security mechanism used today. Reasonably good firewalls can be created with routers alone, but it may be difficult to program the router to implement all

security filtering rules when security requirements are conflicting. Routers can be used as firewalls under certain circumstances, but not in isolation. A second router and a host that serves as a gateway are usually required[7].

Usually, it is quite straightforward to allow or disallow access to or form a system or group of systems. But, filtering on a per-service-type basis can only be done based on the knowledge of the port utilization of the service concerned. for instance, the SMTP used for most electronic mail transport on the Internet always binds to destination port 25 [32] .

If routers are used, the best technique (though not necessarily robust) is either a bastion or a diode firewall configuration.

3.2.1.1 Bastion Firewalls

A bastion firewall (Figure 3.4) is basically a pair of routers with a linking host in the middle. All are configured to protect the secured network from outside access. The external router on the Internet side sends all its data only to the host PC, which in turn communicates with the other router on the secured LAN side. The secured router, likewise, talks only to the host. Thus, the hostile network and the secured network are isolated from each other, and the hacker can't get in.

[7] The idea that a single router, configured to allow certain types of packets to pass through, can be used as a robust firewall is a naive approach to network protection.

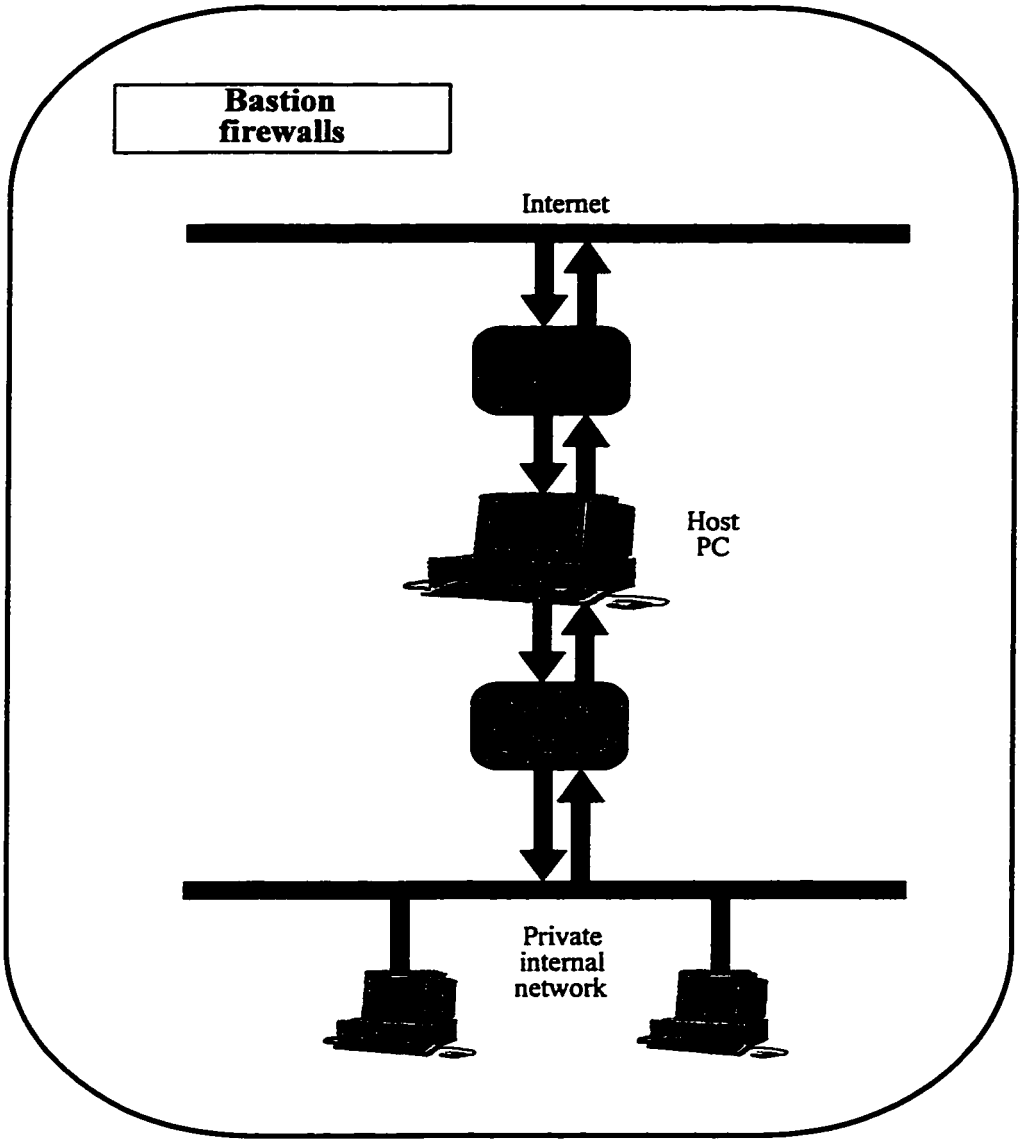


Figure 3.4. Bastion firewalls

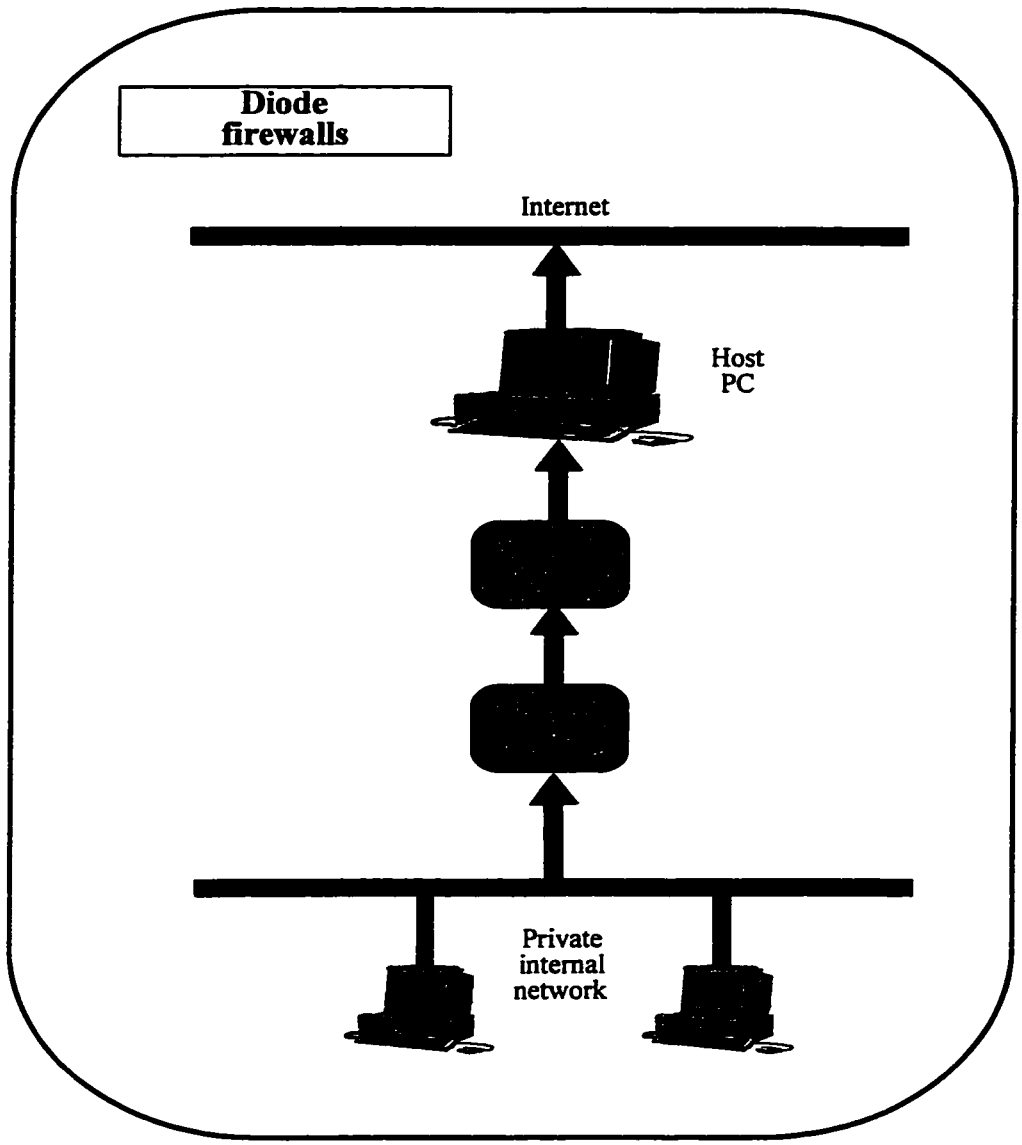


Figure 3.5. Diode firewalls

3.2.1.2 Diode Firewalls

The other workable router-based firewall configuration is the diode firewall (Figure 3.5). This setup is a one-way configuration in which outbound traffic is allowed but inbound traffic is denied. Diodes also consist of two routers and a host, but the host gateway is actually in the hostile network environment (i.e., a PC at the Internet provider's site, for instance). Here, the secured network talks only to the internal router, which talks only to the external router, which talks only to a host gateway. This way, it is fairly straightforward to configure the setup so that the external router (and so, by default the host gateway) is allowed only outbound traffic.

Comparing the two types of router-based firewalls, we can see advantages and disadvantages in each type. The real advantage of the bastion firewall approach is its ability to provide two way communication which has worked well in the past. But hackers and angry or ex-employees who know all the access codes can breach this approach by applying some techniques to forge packet headers. Therefore, the diode setup is considered as a better solution. In diode firewalls, even if an intruder gets past the gateway and into the external router, he has nowhere to go. However, being a one way communication link is a great restriction. Some services, such as the FTP, can not be provided with diode type firewall, because the FTP requires two way connectivity.

3.2.1.3 Steps to Create a Router-Based Firewall

Building a router-based firewall mostly boils down to planning and lots of trial and error. Routing tables and security filters make up the firewall. These rules will allow or prevent users from accessing the Internet gateway or gateways. To achieve more security, an additional router might be considered to sit between the Internet gateway and the internal network. This will add to the complexity of the firewall and access to the Internet, but can help trap intruders. Since the modem/router has its own IP address, which is substantially different from those used in the remainder of the network, an outsider would have to explicitly know and define a path from his system, through the network, through this gateway, and to the internal system. By declaring that no outside initiated traffic may pass through the LAN router (on the other side of the dial-on-demand modem/router), this access can be limited. The steps for building a router-based firewall to protect data from outside access after connecting to the Internet are presented here [39, 40, 41] .

1. The major step is to draw a diagram of the network, highlighting those stations or networks that should be granted access to the Internet or the Internet gateway. This information should be saved for later use in case of disaster recovery and/or system-change processes.

2. Build a table listing the Internet Protocol (IP) addresses associated with each of these networks including the primary addresses and any subnet masking information.

3. Transfer this information onto the network diagram and look for those networks that pass through intermediaries (file servers or other routers) before connecting to the dial-on-demand modem/router.

4. Each of these IP addresses and its path should be declared in the appropriate routing tables and must also be accounted for in security decisions in the firewall. To gain transparent access to the Internet, each router must know about the paths to the others that lead to the Internet gateway. Many routers use router information protocol (RIP) to exchange this information. The router should be checked to see whether the system itself relays this information or if each router must be manually configured. The firewall begins to take shape as we define the relationships between the routers and users and among the routers.

3.2.1.4 Improving the Router-Based Firewall Security

There are steps that can be taken to improve router firewall security [4] :

1. First, never allow in-band programming of a firewall router via telnet. Routers should always be programmed out of band by an RS-232 terminal, and both the router and the terminal should be kept in a restricted-access room.

2. Second, firewall routers, or any devices inside the firewall, should never advertise their presence to outside users. Instead, an external host should be used to advertise the

network to the outside for address purposes, and to handle mail and news groups.

3. Finally, disallow all accesses such as ping, finger, and telnet to the routers and the external host by denying (blocking) certain ports on this equipment[8].

3.2.2 Circuit Gateways

Packet header forging is a fairly well known form of network-layer attack. Subsequently, firewall security of inbound and outbound data traffic is being shifted to higher and higher protocol levels. This involves a more complicated examination of data traffic at the transport, session and application layers [15] .

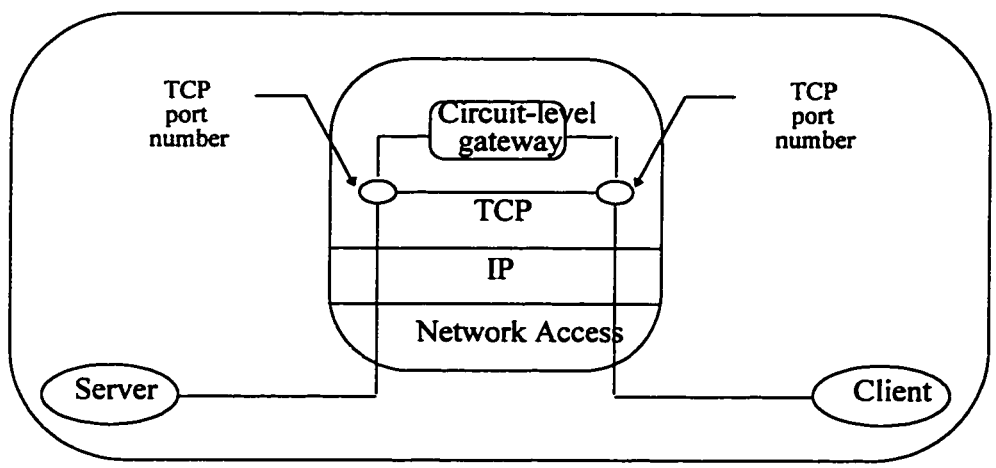


Figure 3.6. Circuit-level gateways

Circuit gateways, as shown in Figure 3.6, provide another way for establishing a firewall. A circuit gateway is similar to the application-level systems, but, while the firewall host computer creates an isolated subnetwork is to sit between the external and

[8] A comprehensive list of vulnerable ports and ways to protect them can be found in Firewalls and Internet Security[33] .

internal networks, it does not interfere in the on going communication after connection establishment. Normally, the circuit firewall isolation network is configured so that both the Internet and the private network can access it to establish a relay connection to the other side.

3.2.3 Application-Level Firewalls

Most firewalls are capable of application-layer processing and usually with support for lower-level filtering. With the combination of these, the effectiveness of a firewall is increased substantially in preventing access by Internet hackers [15] .

An alternative approach to firewall construction is to use a computer rather than a router. This offers many more capabilities, including the ability to log all the activities over the gateway. A firewall system, of a separate, highly secured computer system standing guard over the networks, sometimes called a bastion host, is a critical defense point that must be carefully designed, tightly controlled, and audited regularly.

While a router-based firewall monitors data packets at the IP level, hosts exercise their control at an application level, where traffic can be examined more thoroughly. However, not just any application can be used, it is important to know that the application software and the operating system (that the application software is run on) may have its own security holes.

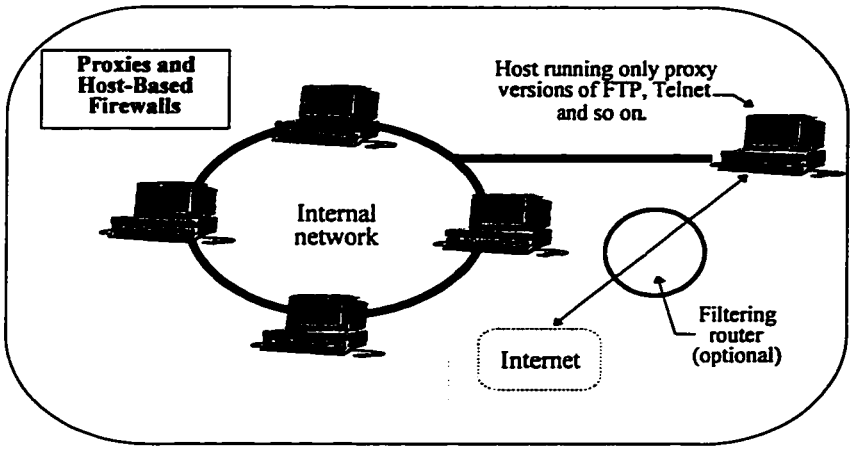


Figure 3.7. Proxies and Host-Based Firewalls

To get around these problems and deal with potentially buggy protocols, application-level firewalls must use specialized software applications gateways and service proxies. However, the cut-down nature of a proxy implies that it can be used only with the application it is designed to serve (Figure 3.7 shows Proxies and Host-Based Firewalls).

The specific path for an application-level firewall varies, but often works in the way shown in Figure 3.8. Packets coming from the Internet pass through the connectivity supplier's network to a router and to the organization's internal network. The arriving packet then goes to a border router that duplicates the function of the external router. The border router, rather than routing packets directly to its destination, it redirects traffic through an external services host and an IP filter host. The external services host runs desired applications such as an e-mail interface. While, the IP choke performs actual protocol filtering. Traffic is then sent through an internal/external service gateway to a separate router on the internal network, configured according to internal security policy.

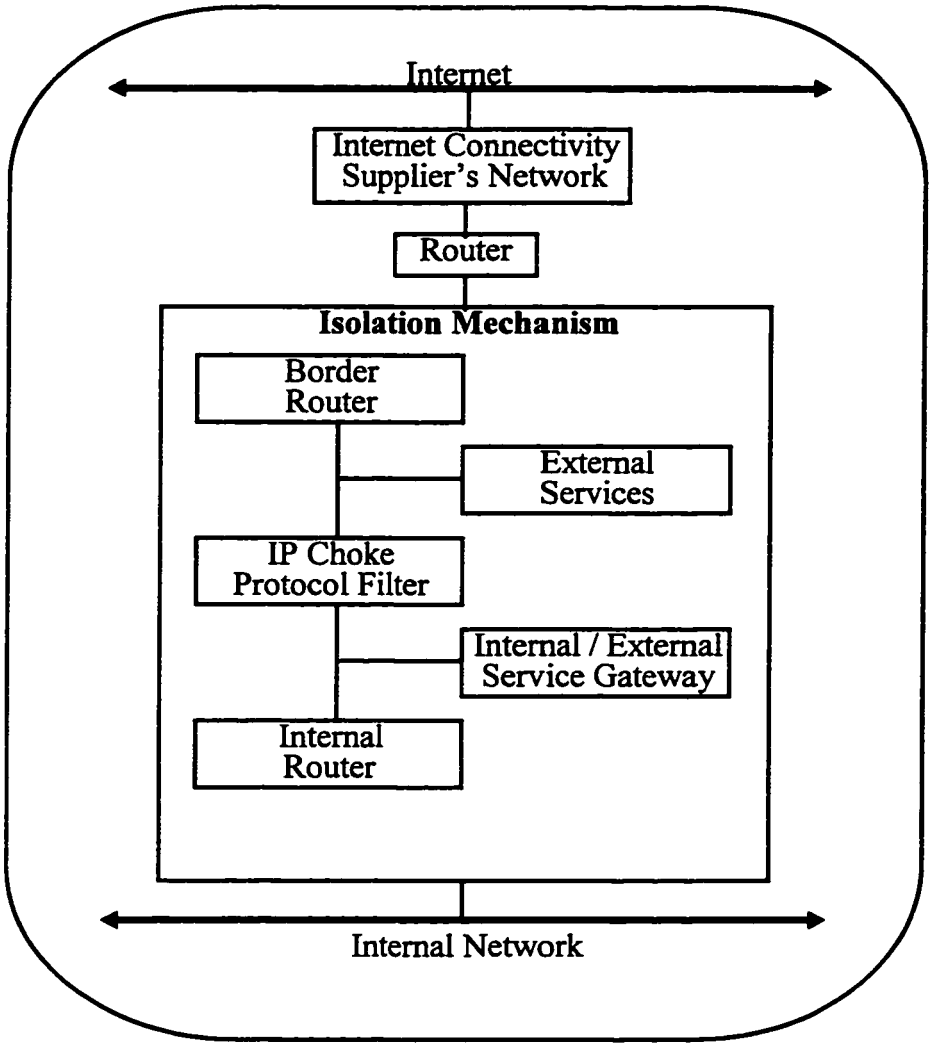


Figure 3.8. Application-Level Firewalls

3.2.4 Comparison of Firewall Types

Firewalls can be a valuable resource when implemented in a security context for host computers, LANs, and Internet connectivity. Firewalls are specifically designed to protect Internet-linked networks against unauthorized access. They work by filtering packets and handling a variety of protocols. Many firewalls provide features such as network address translation, authentication and virtual private networks [14] . Most of these firewalls fall into three classes or firewall types. These types are router-based, circuit-level gateways, and application level gateways. In this section we will discuss firewall classification and compare these firewall types.

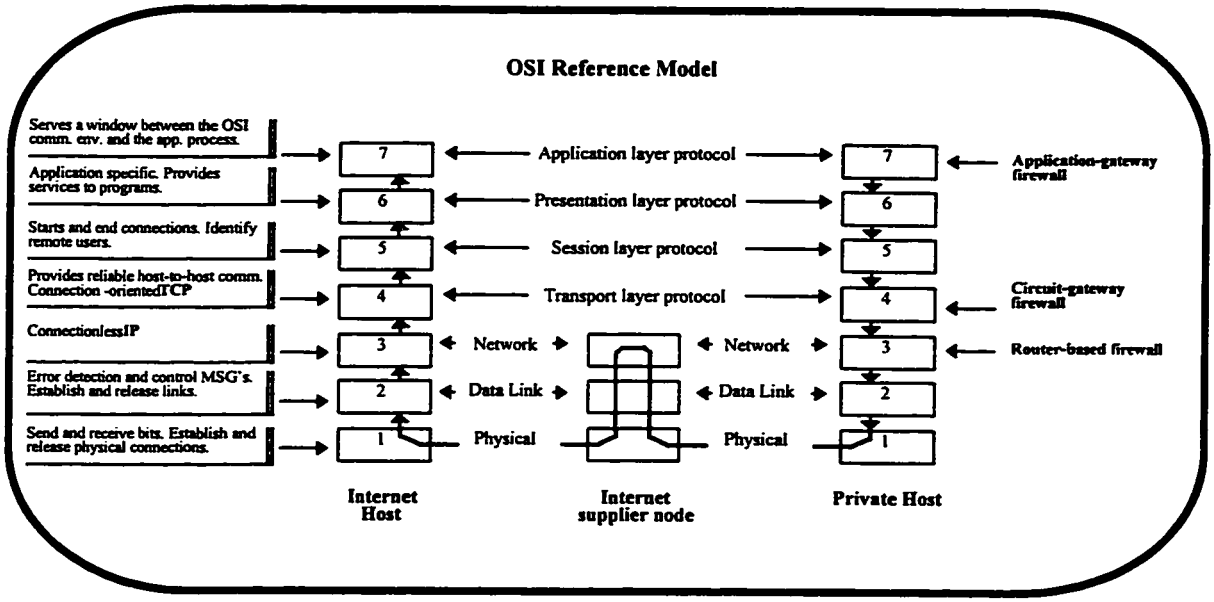


Figure 3.9. OSI Reference Model and Firewall Types

Figure 3.9, relates the three firewall types to layers in the OSI Reference Model. A packet filtering router-based firewall examines each packet to compare its contents against a predefined rules in the security policy. Kernel-level filtering, where packet never leaves the operating system, checks IP addresses and ports. Circuit-level gateways are mostly used for outbound traffic. At a circuit-gateway firewall, users must first authenticate themselves to the firewall, which then opens a new connection to the intended destination. Control is done, only, during the link connection setup time, therefore, applications can do anything the protocol would normally allow them to do. An application gateway is similar to a circuit-gateway, but it can, also, restrict the applications that users can access. This is achieved through a non-privileged user-level program running on a host system to examine all packets sent by a particular application program [38] .

The problem with the router-based approach results from the variety of different protocols that are used on the Internet. Since they only examine one packet at a time, they face difficulties in providing a high level control over such TCP-based protocols as FTP and just about all UDP-based protocols. There are at least three major network services that are not handled well by router-based firewalls, FTP, DNS, and X11. Although most inbound traffic can be denied, FTP is necessary. This is because users inside the secure network need to bring files into the secure network from outside. Therefore, a policy is required to govern the way executables are brought into the network, Such executables should always be checked for viruses and other unacceptable code. There are solutions to this problem,

but they are not always good. Using central authority on top of all the security issues on all network hosts, it is likely that a router, by itself, may be an adequate firewall. However, securing the network still requires an abundance of knowledge about all network hosts' vulnerabilities and regulation of the network traffic that passes through them.

Although most organizations already have basic routers in place that are capable of performing firewall functions, routers, however, have drawbacks. They offer few logging capabilities, are sometimes tricky to configure, do not offer the strong user authentication schemes found in application-level gateways, and are susceptible to spoofing. Therefore, organizations that want more manageability and a highly sensitive interface turn to software-based firewall solutions. The objective is turning off the unwanted traffic without turning off acceptable communications. Firewall products usually can do this more effectively and with less administrator effort than routers, but there is still no guarantee that it will be done correctly. In general, software-based firewalls often provide better logging and authentication and more thorough packet analysis than router-based ones.

Application-level firewalls are software-based firewalls, that are designed specifically to control unauthorized access to the network. These, usually, use a much more conservative security philosophy: "Whatever is not explicitly permitted is denied." They can also handle some of the difficult protocols. Besides providing stronger logging capabilities, many software-based firewalls can provide features like authentication, network address translation, and virtual private networks.

Comparison factors	Firewall types		
	Router-based	Circuit-level gateways	Application-level gateway
Price range ($)	3,000 - 20,000	8,000 - 20,000	8,000 - 100,000
OSI layer	network	transport	application
Security level	low to mode rate	moderate	high
Speed	fast	moderate to fast	moderate
Implementation	easy	easy to mode rate	moderate
Transparency	high	moderate	low
Packet Filtering	Yes	Yes	Yes
Application proxy	No	Yes	Yes
Address translation	No	Yes	Yes

Table 3.3. Firewall types and trade-offs

Application-level gateways running software, that acts as a proxy for Internet services, are generally considered more secure than packet filters or circuit-level gateways because they offer the highest degree of control. An application-level gateway is most appropriate for organizations that have strict security requirements and sufficient resources to implement them. Proxies are used to authenticate and filter transactions between users and host. A proxy agent resides on the firewall and is executed each time a user requests access outside the firewall. The client computer interacts with the proxy on the firewall to relay the transactions. The proxy action can be transparent to the end user, or the administrator may require some authentication from the end user before granting access to a particular service. Some of the disadvantages of using proxies are that they are not available for all services, and they require a lot of attention from the network administrator when new services come online.

Seq. No.	Firewall Products	Firewall Architecture types		
		Router based	Circuit-level gateways	Application-level gateway
1	Black Hole 2.01			x
2	Borderware firewall Server 3.0.1			x
3	Centri NT Firewall			x
4	Connect: Firewall 2.100			x
5	Cyberguard 2.0			x
6	Digital Firewall for Unix 1.0			x
7	Eagle 3.0			x
8	Firewall-1 1.2			x
9	Gauntlet 3.0			x
10	Interceptor Firewall System v2.0			x
11	Interlock 3.0			x
12	Mazama			x
13	NetSP Secure Network Gateway 2.0			x
14	Network-1 1.0-4			x
15	Primstone v2.21			x
16	Private Internet Exchange 2.5.17			x
17	Secureconnect 1.7			x
18	Security Router	x		
19	Sidewinder 2.0			x
20	Silicon Graphics Gauntlet 3.0 firewall for IRIX			x
21	SmartWall			x
22	SOCKS		x	
23	TIS FWTK			x
24	Turnstyle Firewall System 1.1			x
Totals		**1**	**1**	**22**

Table 3.4. Firewall product type classification

Table 3.3 compares different firewall types in terms of their prices, speed, ease of implementation, and so fourth. These distinctions should help in understanding some important differences between the types of network security devices. When considering the cost of a router and a firewall like Firewall-1, the benefits of carefully evaluating which option best suits the needs are obvious. Firewall products, as explained earlier, can be classified in three distinct architectural types. Product classification is shown in table 3.4, where, it appears that most of them fall in the category of application-level gateways.

3.3 Firewall Architectures

As mentioned earlier, firewalls provide the best protection mechanism for network security against intrusion and unauthorized practices. However, firewalls can be implemented in a number of ways in order for the different security requirements and policies to be accommodated using different firewall architectures. In this section, we will discuss how network security can be achieved with different firewall architectures, then we will briefly present some alternative solutions and strategies for a secure Internet access.

3.3.1 Alternative Solutions and Strategies for a Secure Internet Access

Several solutions and strategies can be used in configuring Internet connections so that unwanted intrusion can be prevented. Each of these strategies has its own advantages and disadvantages. Some of these solutions include the use of two sets of hosts (secure and non-secure), the use of a firewall host, the use of router filtering, and the use of some firewall packages [42] .

3.3.1.1 Using Two Sets of Hosts

This alternative is the most simple and obvious method that can be used to provide a completely secure environment where two sets of hosts: secure and non-secure sets are used. The set of secure hosts are not connected to the Internet and are connected only through an isolated network. On the other hand, non-secure hosts are a set of computers

that are connected to the Internet. The non-secure hosts can not communicate directly with the secure hosts, therefore the communication must be performed manually, such as through a tape. Although, security maintenance can be minimal because there is no critical or vital information on the non-secure hosts, this method has the disadvantage of being untimely and annoying to users.

3.3.1.2 Using a Firewall Host with User Accounts

A slightly more convenient and secure access to the Internet can be achieved via a firewall host which doesn't route traffic, therefore no traffic is allowed to pass through while both incoming and outgoing connections are allowed. Users who are interested in accessing the Internet must login to their accounts on the firewall host. So, if a user wants to transfer a file from one host to another, he should first transfer the file from the source to the firewall host and then login to the firewall and transfer the file to the destination host. The advantage of such implementation is that security intrusions are limited to a single point of access. However, it is still not very convenient to users and also maintaining the security can be a difficult task with the increase in the number of users requiring access to this host.

3.3.1.3 Using Router Filtering

In place of the firewall host. a router can be used to filter packets based on their source/destination host and port addresses in order to provide a secure Internet access.

Router filtering can be used to prohibit inbound traffic to low numbered TCP ports, for instance less than 1024[9], while allowing all outbound traffic. Under this solution, users can gain access to Internet services directly from their workstations, while prohibiting unwanted external access. Although, this solution is more convenient to users, it has a major problem because when the routers security is compromised, the security of the whole network is compromised.

3.3.1.4 Using Firewall Packages

None of the above methods seem to be ideal, therefore some packages, such as SOCKS [42] , have been developed in an attempt to provide the best features of these solutions, while reducing security problems and maintenance to a minimum. This method provides more convenient connectivity to Internet users, even though they are not required to have accounts on the firewall host. In addition to limiting the possibility of security intrusions to a single point of direct Internet connectivity, much more security is achieved. Therefore, using firewall packages provides a better mechanism for securing Internet connectivity as well as more secure and convenient access method to the local network in general.

[9] Ports less than 1024 are reserved for well-known network services such as finger, ftp, and telnet. On the other hand, ports greater than 1024 are allocated as needed by the UNIX operating system and this is generally where outbound port numbers are obtained.

3.3.2 Achieving Network Security Via Various Firewall Architectures

Router-based and host-based firewall types can be used in a number of ways to achieve network security. Mainly, there are three forms of firewall architectures with varying costs and levels of attained security. The three architectural forms are: Dual-Homed Host, Screened Host, and Screened Subnet architectures. The simplest firewall configuration would consist of a dual-homed gateway, in which a workstation with two network interfaces is connected to both networks with IP forwarding disabled. In addition there exists a number of variations of these firewall architectures. In this section we will only present the three forms of main firewall architectures.

3.3.2.1 Dual-Homed Host Architecture

A Dual Homed Host or gateway is a firewall system consisting of a workstation with two network interface cards to link two networks together, while not allowing traffic to flow directly across it. The dual-homed host is, by definition, a bastion host, where it can be reached from both the private network and the Internet. In some sites, the dual-homed host is used as the base for user operations, where users are required to successfully login to the dual-homed host before they can login to other hosts on the Internet, or perform other tasks such as FTP [43] .

The critical aspect of dual-homed firewall hosts is that direct routing of packets between network interface cards is disabled at the network layer level. Therefore, data transfer

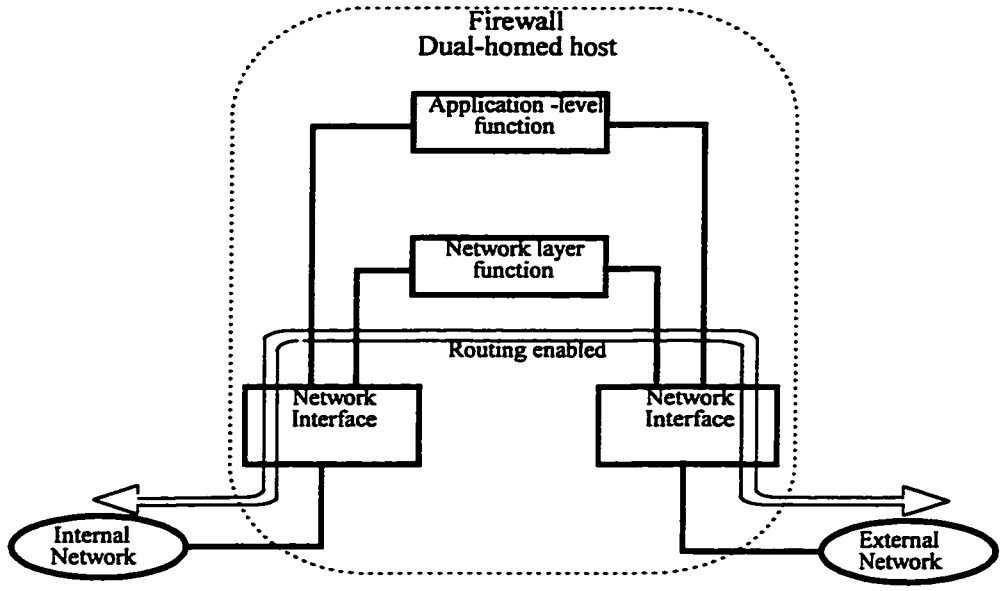


Figure 3.10. A misconfigured dual-homed firewall.

and exchange of information take place at the application layer under the control of the appropriate application. As shown in Figure 3.10, enabling routing at the network layer level results in a misconfigured dual-homed firewall. When this happens, all packets will be directly forwarded from one network interface card to another, therefore, bypassing all application level filtering rules of the firewall running on the dual-homed host [44] .

Because, the option to route services at a network level in dual-homed gateways is generally not available, dual-homed hosts are not flexible compared to other firewalls where hosts and routers are combined together. However, with a dual-homed gateway, since routers are not an integral part of the security system, network administrators can be more confident that network traffic will not be able to somehow leak through [5] . However, in practice, the dual-homed host architecture is liable to failures, and when this happens, packets will cross

from external to internal network. This type of failure is totally unexpected, therefore, it is unlikely to have complete protection against attacks of this kind [43] .

3.3.2.2 Screened Host Architectures

The screened host gateways are the most common and most flexible form of firewall architectures. The screened host architecture provides services from a host attached to only the internal network. In this architecture, the primary security is provided by a packet filtering router, that is used to block all traffic between the Internet and all hosts on the private network, with the exception to one bastion host. It is also possible to configure the screening router so that the Internet can be reached directly from some nodes on the private network via network services such as TELNET and FTP.

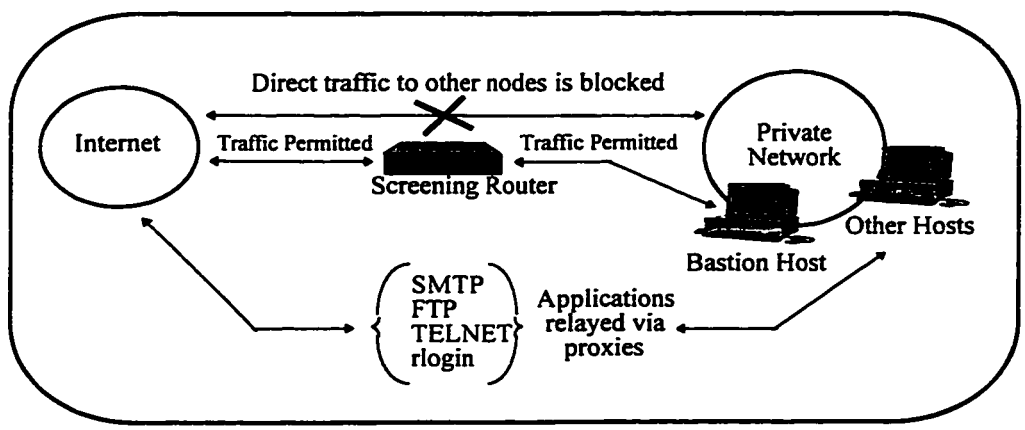


Figure 3.11. Screened-Host firewall architecture.

Figure 3.11 represents the architecture of an installed screened host gateway in which a router and a firewall bastion host are combined. In this case, the implementation of security

controls is shared between the router and the bastion host. While the router controls network-level access, the bastion host provides application-level control [5] .

The bastion host must maintain a high level of security, because all Internet connections are opened on the bastion host. Whereas any external system trying to access internal systems or services must connect to the bastion host, only certain types of Internet connections are allowed to provide services such as incoming e-mail. On the other hand, packet filtering permits the bastion host to only open external connections as allowed by the site's particular security policy. The screening router can use packet filtering configuration to perform one of the following [43] :

- Allow internal hosts other than the bastion host to open connections to the Internet hosts for certain internal. This can be made possible by allowing the desired services via the packet filtering.
- Force internal hosts to use the proxy services via the bastion host, by preventing any direct external connection from internal hosts.
- Mix and match, depending on the particular sites policy, these approaches for different services. Where, some may be allowed directly via packet filtering, while others may be allowed only indirectly via proxy servers.

The screened host architecture may seem more risky than a dual-homed host architecture. Because, in the screened host architecture, it is allowed to move packets

from the Internet to the internal networks, while external packets are not allowed to reach the internal network in a dual-homed host architecture. However, for most purposes, the screened host architecture provides better security and usability than the dual-homed host architecture, because defending a router is easier than defending a host which is also liable to failures. Some of the major disadvantages that are found in this architecture include [43] :

- lack of any protection between the bastion host and other internal hosts to provide another level of network security in case an attacker manages to break in to the bastion host.
- because the router presents a single point of failure, the entire network is exposed to attackers once the router is compromised.

3.3.2.3 Screened Subnet Architecture

Because of the disadvantages that are found in the screened-host architecture, the screened subnet architecture has become increasingly popular. In this architecture, a screened subnet is placed between the Internet and the private network with the intention to effectively hide the private network from the Internet and make it invisible to outside users. When using screened subnet architecture, both the Internet and the private network can communicate through nodes on the screened subnet, but they can not communicate directly.

The screened subnet architecture is designed by adding a perimeter network to the screened host architecture. It adds an extra layer of security as well as further isolate the internal network from the Internet. Despite the best efforts to protect them, bastion hosts are the most vulnerable machines on the network. But unlike the screened host architecture, when someone successfully breaks into the bastion host, the additional subnet hides the internal network and the hacker will be caught in the middle [43] .

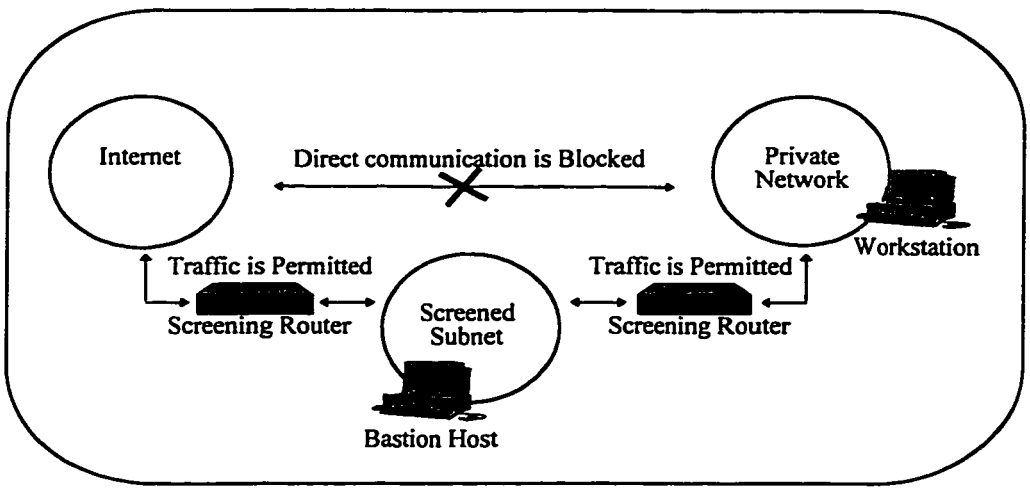


Figure 3.12. A Screened Subnet firewall.

As can be seen in Figure 3.12, the simplest type of screened subnet architecture consists of two screening routers connected to a perimeter network. One of these routers is placed between the internal and the perimeter networks, while the other is located between the external network (usually the Internet) and the perimeter network. With this type of architecture, there is no single vulnerable point that will compromise the security of the internal network. If an attacker somehow breaks into the bastion host, he would still have

to get past the interior router. Therefore, to break into the internal network, it would be necessary to get past both routers as well as the bastion host.

Layered series of perimeter nets can be placed between the outside world and the internal network, with less trusted and more vulnerable services located on the outer most perimeter networks. In this approach, if an attacker breaks into a machine on an outer perimeter network, he will have a harder time successfully attacking the internal machines because of the additional layers of security. But this can be only true if the different layers are meaningful, otherwise, if the same filtering system is used between all layers, then no additional security is provided by the additional layers.

3.3.2.4 Variations of Firewall Architectures

Beside the three most common firewall architectures mentioned in the previous sections, there exists a number of various firewall architectures which are variations of the three most common firewall architectures. These include [43] :

- Using multiple exterior screening routers.
- Using multiple bastion hosts. One for each service or group of services.
- Using dual-homed hosts and screened subnets.
- Using multiple perimeter networks.

- Merging of interior and exterior screening routers.
- Merging of bastion host and the exterior screening router.

However it is dangerous to use the following as valid variations of firewall architectures [43] :

- Using multiple interior screening routers: One reason behind this that internal traffic may be routed through the screened subnet. In this case, intruders who managed to get into the screened subnet, but not into the protected network will be able to read the routed internal traffic and learn valuable information.
- Merging of the bastion host and the interior screening router: In this case, when the bastion host is compromised, there will be no more protection to the internal network.

3.4 Firewalls as Part of the Overall Security Policy

The focus on firewalls has been concentrated mainly on the layout of various possible configurations of routers, host systems, interfaces and sub-nets. But it is essential to think of a firewall as a part of an overall security policy. While the security policy defines the services and access to be permitted, a firewall is an approach to security by helping in the implementation of a larger security policy [33] . In its general definition, a firewall is a system or group of systems that enforces access control policies between two networks. While, more precisely, it is a system or a collection of components placed between two

networks to protect the trusted network from the untrusted network by maintaining the following three properties [3] :

1. all traffic going out from the inside and all traffic coming in from the outside, must pass through the firewall.
2. only authorized traffic, as defined by the local security policy, is allowed to pass through the firewall.
3. the firewall system itself is immune to penetration.

Many of the commercially-available firewall products are very powerful and by deploying a firewall, most of the security problems that are associated with internetworking can be fixed. For organizations having, or planning to create a connection between their network and another network, firewalls are considered as the best defense line. However, firewalls, at least in the narrow sense of the term, do not provide complete solution [3] . A firewall should constitute of both the policy and the implementation of that policy in terms of [33] :

1. network configuration,
2. hosts systems and routers,
3. other advanced security measures such as authentication instead of static passwords.

3.4.1 Functional Requirements

Firewalls should include certain security measures to protect private networks when connected to the Internet. These measures should include an integrated rights check that permits access control to the Internet based on the: user name, calling client workstation, and requested service. In general, network secure connection to the Internet requirements should include the following [29] :

1. **Authentication methods of different strengths:** Different authentication methods should be supported by the firewall system. These methods may range from "TCP reserved port authentication" to the deployment of chip cards, depending on the area of usage.

2. **Access control on the basis of service, host, and user:** Not all users in the LAN should be granted access to all services of the Internet. Rather, only certain users should be enabled to use distinguished services.

3. **Restriction of services:** Blocking parts of the services, should be possible as well as preventing the usage of certain destination addresses.

4. **Logging for error diagnosis and accounting:** All error situations, potential assaults, destinations of connection establishments, and the size of the data transferred, should be logged by the firewall system

5. **Protection of the firewall server against the Internet service access system:** It should

be impossible for a user of the Internet service access system such as firewalls or proxies to threaten the security of the host on which the server is running.

Usually security requirements and functional requirements are contradicting objectives. Nevertheless, when using firewalls to achieve network security, firewalls should satisfy the following functional requirements:

1. **Support of as many communication protocols as possible**: Since, it may be difficult to cover many different protocol families such as DECnet., AppleTalk, and TCP/IP, at least one protocol family with as many sub-protocols as possible should be supported.
2. **Minimal modification effort:** In order to implement an Internet service access system, network programs generally have to be modified. however, such modification effort should be kept as low as possible.
3. **Low resource utilization on the firewall server:** Resource requirements, such as main memory, disk space, and CPU time, that is needed to run the firewall system components on the firewall server should be as minimum as possible.

3.4.2 Defining Firewall Specifications

The next step after deciding to use firewalls protection method to implement the security policy, is the acquisition of a cost-effective firewall that renders a suitable level of protection. What firewall is right for a given customer?. The answer to this question, is that there is no

single firewall solution for all customers. This is because security needs are diverse such as: the level of needed customization, the available level of experience with UNIX, whether integrated Internet services are required or not, and the size and complexity of the network. The exact features of a firewall to provide effective implementation for a specific security policy can not be stated here, but a firewall in general should [3, 6] :

- Support the security policy as well as a design policy that deny all services except those specifically permitted, even if it is not the policy in use.
- Be able to accommodate new services and requirements, as the security policy of the organization changes.
- Support advanced authentication procedures, or at least, it should be integrable with other advanced authentication procedures.
- Control service access to specified host systems as needed by means of filtering techniques.
- Support the use of a flexible and user friendly IP filtering language that is capable of filtering on as many attributes as possible. Such attributes should include IP addresses of both the source and the destination, the type of protocol, TCP/UDP ports for the source and destination, as well as inbound and outbound interface.
- Use of proxies for services such as the FTP and TELNET, X and gopher, in order to use

centralized advanced authentication measures at the firewall.

- Support centralized handling of site e-mail in order to reduce direct SMTP connections between the site and remote systems.
- Distinguish between the protected public information servers and other site systems that are not allowed for public access.
- Have the capability to centralize and filter dial-in access.
- Contain logging mechanisms of traffic and doubtful activities, as well as mechanisms for the reduction of log records in order to make them readable and understandable.
- Ensure firewall host integrity by implementing a secured version of the operating system as a part of the firewall, if it is required to have an operating system such as UNIX. The operating system should have all patches installed with other security tools as necessary.
- Be simple in design so that it can be understood, maintained, and verified for its strength and correctness.

In addition, the firewall as well as any related operating system should be, in a timely manner, updated with patches and other bug fixes. Especially that the Internet is changing continually, new vulnerabilities will always arise. Therefore, it is important that the firewall is flexible in order to adapt to changing needs, as well as staying current on new threats and vulnerabilities.

Comparison Factors	In-House Firewall	Vendor-Supplied Firewall
Resources	The organization should consider whether it has the internal resources to build and maintain the firewall.	The vendor is responsible for the necessary resources to build and maintain the firewall.
Experience	Enables personnel to understand the specifics of the design and use of the firewall.	In-house personnel are not given the opportunity to understand the specifics of the design and use of the firewall.
Overall cost of the firewall should consider: (This is in addition to the cost of equipment)	The time required: 1. To build and document the firewall. 2. To maintain the firewall 3. To add new features as needed.	1. Free installation. 2. Free maintenance services if provided by the vendor. 3. The extra charges when adding new custom features.
Maintenance	Organization's responsibility	Usually, Vendor's responsibility
New features	Can be custom tailored.	Usually, are general purpose.

Table 3.5. Tradeoffs between in-house and vendor-supplied firewalls

3.5 Buy or Build a firewall

The decision whether to buy or build a firewall, involves some advantages and disadvantages as shown in Table 3.5. For organizations that have the capability to develop a firewall from scratch or put it together from available equipment and software components, it will be to their advantage to build their own firewall. On the other hand, for other organizations, a wide range of firewall service technologies are offered by several vendors. These include: the necessary hardware and software, the development of security policy, security reviews and risk assessments, as well as security training. Whether the decision is made to buy or build a firewall, security policy has to be developed first.

One very important factor to consider when computing the cost of a firewall, is the cost of its administration. In addition to the initial required resources during the building phase of the firewall, an organization should decide whether or not it has the resources for operating

and maintaining a successful firewall. To help in making this decision, the organization should consider the necessary resources to handle issues such as: firewall verification for correctness and expected performance, maintenance, enhancements, updates, backups, and training.

3.6 Future Firewalls

An emerging trend is the integration of firewall categories, serving the diverse needs of an organization. Additional value-added features that are expected to become more significant in emerging next-generation firewalls include: encryption, improved ease-of-use, complete transparency, enhanced auditing and intrusion detection, and Internet access capabilities. And looking ahead, we see five major trends that will bring change to the world of firewalls [45] .

1. **SMALLER NETWORKS**: The networks that firewalls protect are expected to be smaller, on the average, as use of the Internet extends to the sections of small and mid-sized organizations. This will lead to cheaper and easier to use products. There are products designed for medium-sized business that come with built-in Internet application servers, such as Web and mail servers, to further reduce the likelihood of an error causing a security compromise. Another firewall. is the DOS-based Network-1 that is relatively inexpensive, and if someone breaks into it, there's nowhere for him to go because it doesn't have a TCP/IP stack. Building a Windows NT version of firewall

products is one of the most important projects of most firewall companies. The NT operating system makes the workgroup-level firewall far more useful, but there is still concern about the inherent security of NT. Eventually, there will be more demand for personal firewalls.

2. **NEW PROTOCOLS:** The Internet Engineering Task Force has approved a new version of the Internet Protocol, IPng or IPv6. This will brings two changes to the firewall community:

   a. First, it increases the need for address translation, because most companies do not like to redo their internal networks.

   b. Second, the protocol itself contains provisions for encryption and authentication, so those features will not be as important in firewalls.

3. **THE ATM CHALLENGE:** The cells that Asynchronous Transfer Mode networks ATM uses to transmit data are often smaller than the size of an IP packet. Therefore it is impossible for firewalls to examine a network address that is split between two cells. Some vendors have solved cells problem by putting two ATM cards in a router to merge cells for a quick security trick, then re-segment them, and let them continue on their way again as ATM data. Network security will be affected intensively by the end-to-end nature of ATM addressing. Network Systems has proposed an ATM firewall that would compare ATM cells with a security policy and discard those that do not match.

4. **HARDWARE VS. SOFTWARE:** Firewalls forward packets, therefore, the task of firewalls can be accelerated by specialized silicon integrated in routers and other internetworking equipment. For instance, Sun's Sunscreen product is an encrypting router and firewall built on a SPARC platform. This gives an indication that firewalls may end up as a hardware business [46] .

3.7 Summary

In this chapter, firewalls were studied in more details. At first, an overview of firewalls was given, then, more detailed discussion of firewall types and architectures was presented. Next, Firewalls as part of the overall security policy and whether to buy or build a firewall were discussed. Finally, future directions of firewalls were investigated.

CHAPTER 4

FIREWALL EXAMPLES

Vendors and researchers are trying to find methods for granting the users in the LAN access to the services and resources of the Internet without decreasing the security of the LAN or the firewall server. This objective is achieved with varying degrees of success by using firewall servers to connect local area networks to the Internet.

In this section we will explain some existing firewall products: (1) Digital's Three Way Isolation (2) The Internet service access system IpAccess developed at the European Institute for System Security (E.I.S.S.), (3) the Stanford University Research Firewall (SURF), a network firewall design that is suitable for a research environment, (4) the TIS firewall toolkit implemented by Trusted Information Systems Inc., and (5) the SOCKS library developed by D. Koblas.

4.1 Digital's Three Way Isolation

An example of the third type of firewalls (the application-level gateway) is the "Digital's Three-Way Isolation", as shown in Figure 4.1. It requires three computers, Gatekeeper, Gate (circuit gateway), and Mailgate. Gatekeeper resides on the external network, Mailgate resides on the network to be protected, and Gate resides on both. In this

way, a screened subnet is established that isolates private system from the Internet or any other public system. The screening software runs on Gate, a secure host. There are no user accounts (only system administration accounts) on any of the hosts, and the applications loaded are customized Unix utilities to pass acceptable packets back and forth over the link.

Gatekeeper is the doorway to the outside world. It is the root DNS (Domain Name System) server of the system for the Internet. It is also where the applications gateways or proxies reside. Gatekeeper should be configured to accept logins only from trusted hosts, and the packets from these are screened according to the established security policies. It records all login attempts, and can be programmed to send an alert to the systems administrator in the case of repeated unsuccessful attempts. If Gatekeeper is compromised, damage is limited to that single system, because all security screening policies of the network are stored on Gate which does not accept logins from external systems. Therefore an intruder can not get into Gate and change network security screening rules.

Firewall Service uses a sendmail proxy for the E-mail to pass messages across the firewall. All mail between internal and external addresses is routed through Gatekeeper. If incoming mail is destined for a host on an internal network running TCP/IP, then Gatekeeper forwards the mail (through Gate) to that host. If mail is destined for a host that's not running TCP/IP, then Gatekeeper forwards it to Mailgate, which serves as a gateway to other protocols. Outgoing mail is forwarded through Mailgate and then on to Gatekeeper. Mail destined for another internal address never leaves the internal net.

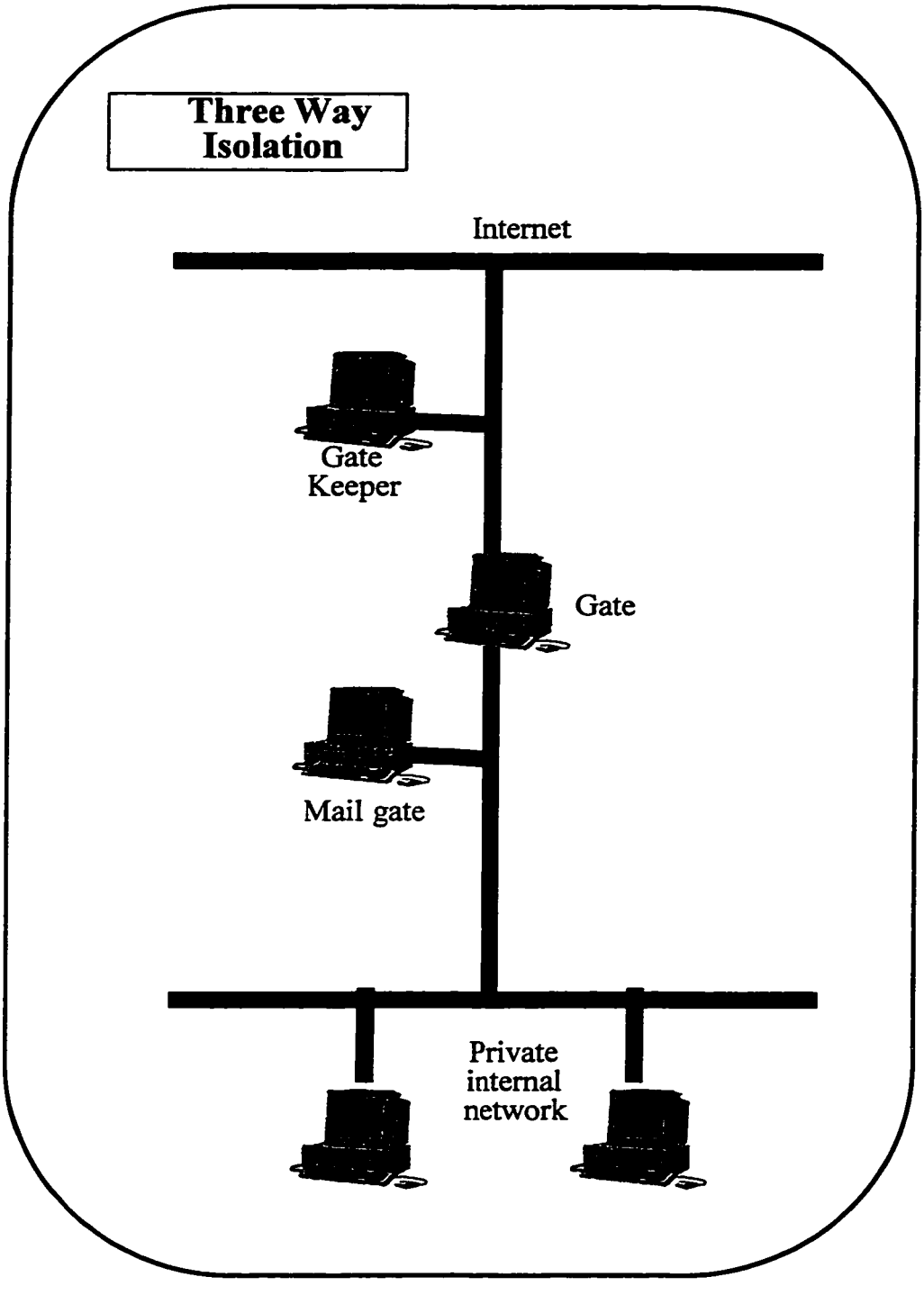


Figure 4.1. Digital's Three Way Isolation

Besides maintaining the system's screening policies, Gate logs all attempts to connect with the internal hosts, as well as all successful log-ins to Gatekeeper and any requests for remote connections across the firewall. Alarm parameters can be set to inform the systems administrator of any problem. Optionally, Gatekeeper can be configured to require hand-held authentication tokens, for successful log-in.

Firewall Service uses a sendmail proxy for the E-mail to pass messages across the firewall. All mail between internal and external addresses is routed through Gatekeeper. If incoming mail is destined for a host on an internal network running TCP/IP, then Gatekeeper forwards the mail (through Gate) to that host. If mail is destined for a host that's not running TCP/IP, then Gatekeeper forwards it to Mailgate, which serves as a gateway to other protocols. Outgoing mail is forwarded through Mailgate and then on to Gatekeeper. Mail destined for another internal address never leaves the internal net.

4.2 IpAccess Firewall

The Internet service access system IpAccess [29] belongs to the corporate network security environment category. The IpAccess firewall system uses an integrated rights check that permits access control to the Internet based on the: (1) user name, (2) calling client workstation, and (3) requested service, while, for reasons of security, it does not allow users to login on the firewall server. Therefore, a special Internet service access system (IpAccess),

is used. IpAccess uses two different methods to grant the users in the LAN access to the services and resources of the Internet:

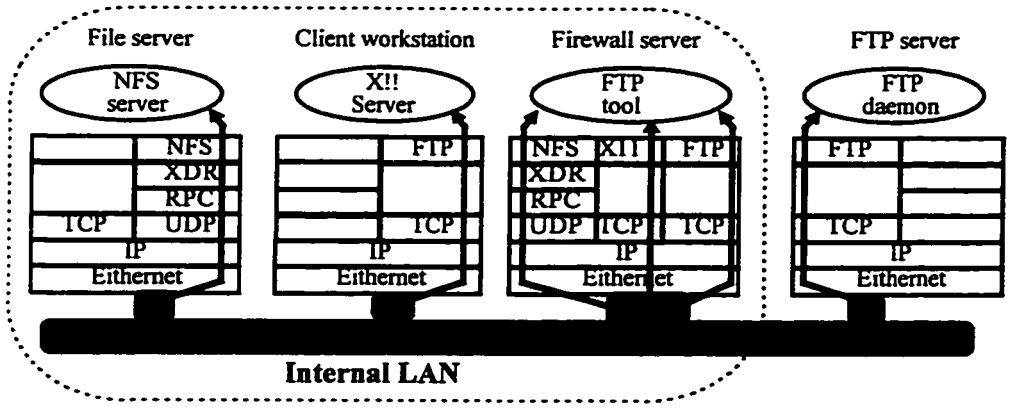


Figure 4.2. Application forwarding in the TCP/IP protocol stack

1. **Application forwarding:** The method shown in Figure 4.2 is called application forwarding, because the protocol conversion is performed on the application layer of the TCP/IP protocol stack. In this method, the network applications run on the firewall server itself and communicate for instance via X11 with the user and via FTP with the Internet. Therefore, the network programs allow access to any network protocol that is available on the firewall server machine especially UDP and raw IP.

2. **TCP forwarding:** The second method is called TCP forwarding because the Internet service access system just copies TCP packets without caring about their contents. In this method, service access is performed on the transport layer as shown in Figure 4.3.

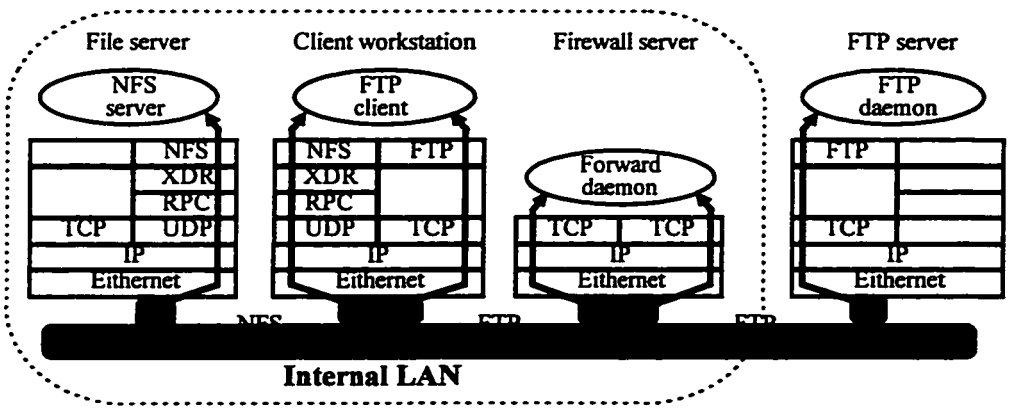


Figure 4.3. TCP forwarding in the TCP/IP protocol stack

Both methods are used by the IpAccess Internet. service access system, because certain network services can only be realized with either of the two methods. The service program itself logs the actions of the user. In IpAccess, the connection through the firewall in general does not take place transparently and the firewall server results in a bottleneck for assaults from the internet into the LAN and allowed communication between the LAN and the Internet.

4.2.1 Internet service access on the application layer using application forwarding

IpAccess works as an Internet service access system on the application layer (application forwarding). Alternatively, after adaptation of the applications in the internal LAN, one of these applications allows the Internet service access on the TCP level. IpAccess allows granting Internet access on the basis of user and service. The service programs running on the firewall server do not decrease the security of the machine.

In the application forwarding, network applications are running on the firewall server itself in a container environment. The user communicates with the service via the X11 protocol and the X server on the client workstation. Protocol conversion is performed on the application layer of the TCP/IP protocol stack. The service can establish connections into the Internet since it is running on the firewall server, where, deployment of all network protocols that are supported by the firewall server is possible. The container environment assigned to the user is set as the root of the file system. The service can be run in the standard Unix file system using the service privilege "nochroot". The service program itself also logs some actions such as destination of a connection establishment, amount of data transferred, duration of connection, or error states. Using This information an error analysis as well as accounting of net access costs is possible.

4.2.1.1 Initiation of the Internet service access

In the application forwarding, the connection establishment to the IpAccess daemon on the firewall server is performed explicitly by the user. Figure 4.4 shows the Course of the service initiation from the request by the user on a system in the internal LAN ("client workstation") to the termination of the service.

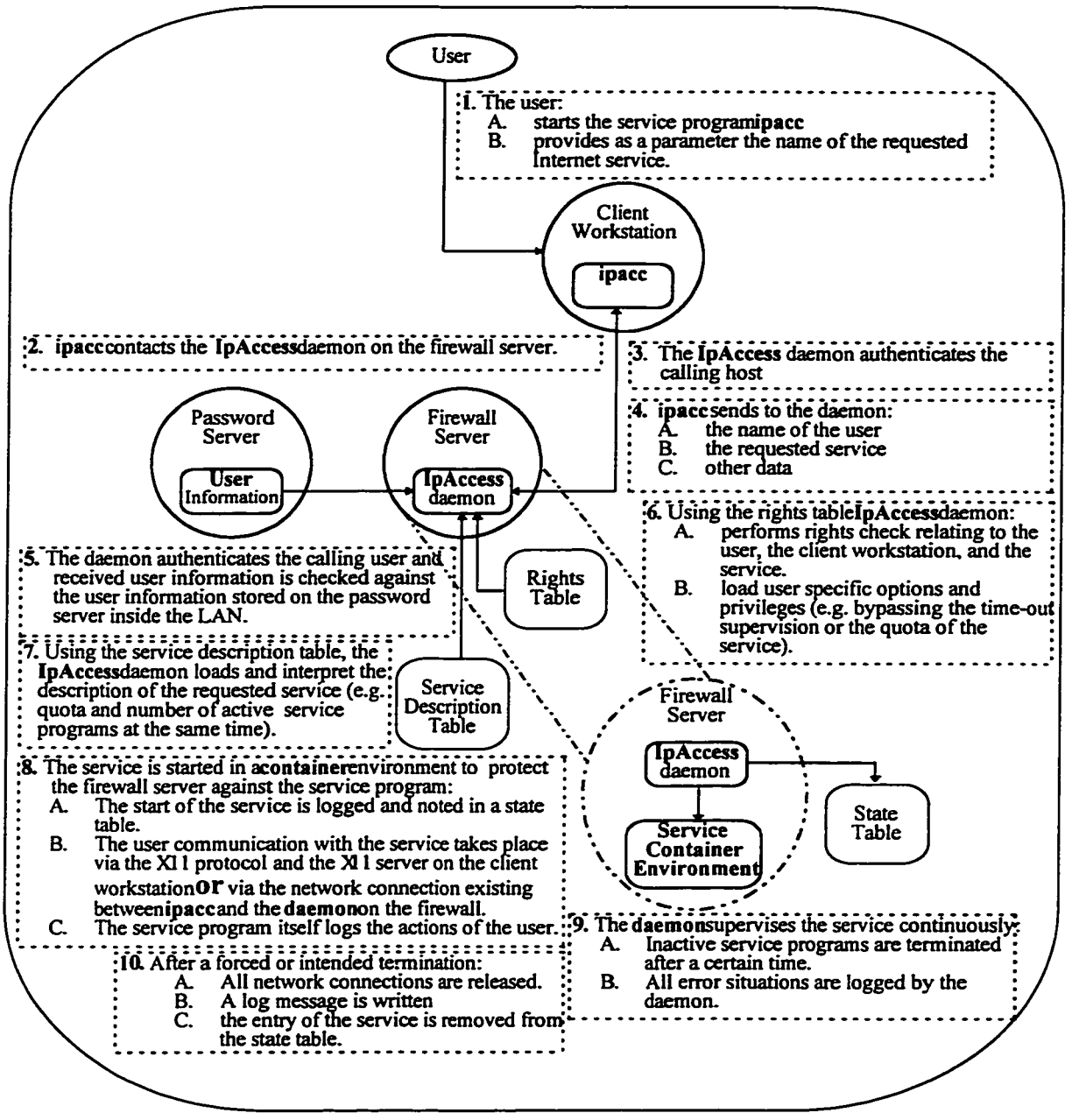


Figure 4.4. Initiation of the Internet service access for the application forwarding

Service establishment and host authentication:

When IpAccess contacts the daemon on the firewall server, The calling client workstation is first authenticated by the daemon to ensure that no system in the outside Internet is able to use the IpAccess service on the firewall machine. To perform the rights check later, this identity of the calling client workstation is also needed.

A variety of methods can be used for host authentication, ranging from a simple check of the Internet address of the client workstation[10] to authentication system SELANE and Unix / vendor-specific systems like "Secure RPC" or "Kerberos".

User authentication:

After a successful authentication of the client workstation, ipacc (IpAccess service startup program) sends to the IpAccess daemon:

1. the name of the calling user,
2. the requested service,
3. the XII display to be used for the interaction with the user, and
4. service specific parameters.

[10] To make this work, a router has to be configured such that from the Internet no IP packets with sender addresses from hosts in the Internal LAN can reach the firewall server

Different authentication methods of the calling user can be applied. These include methods that are based on cryptographic algorithms. The simplest method is the "TCP reserved port. Authentication"[11]. In this case, the daemon trusts the information of ipacc after verifying that the client workstation is trustworthy and that the calling process has root privileges. In addition, the information reported by ipacc: the user's name, home directory, UID etc. can be compared with data received from a password server[12]. This eliminates the need for storing user information on the firewall sever.

4.2.2 Internet service access on the TCP layer using TCP forwarding

When using the Internet service access on the application layer level, only a small program is necessary on the client workstation to start the application on the firewall server. On the other hand, using an Internet service access on the transport layer TCP, the application is running on the client workstation, where it is started by the user. Where, the user communicates with the service via the network connection existing between the IpAccess and the daemon on the firewall server. In this case, the firewall server just forwards TCP packets between LAN and Internet without caring about their contents. There are two distinguished phases when executing an application using the Internet service access on the transport layer:

1. In the **startup phase** a control connection to the IpAccess daemon on the firewall server

[11] "TCP reserved port. Authentication" is also used by the Unix programs rlogin and rsh.
[12] The password server is the host in the LAN holding the users' password file.

is established. In this phase, mainly the methods described in section 1 for the start of an application on the firewall server are utilized.

2. In the **forwarding phase** all TCP/IP network connections to external systems are detoured to the daemon on the firewall server. Information is transmitted over the control connection established at startup time as the actual destination address of a connection establishment.

4.2.3 Comparison of Application and TCP forwarding

The choice of which of the two Internet service access methods is to be used, depends on the environment, because, both methods have their strengths and weaknesses. It is also possible to employ both methods in parallel to take advantage of the strengths of both of them. The following comparison shows the strengths and weaknesses of both methods:

4.2.3.1 Application forwarding

+ All protocols of the firewall server can be used.

+ Because the user is forced to use the network service programs within the respective container environment, it becomes possible to restrict transfer direction and distention addresses of services.

- The firewall server has to provide the resources for the data representation and the

graphical user interface, since all applications are running on it.

- From the firewall server, it is impossible to access special hardware of the client workstation such as the audio and the video hardware.

- The applications running on the firewall server should be able to access files of the user via NFS or another equivalent network wide file system. But, this may be impractical in large nets with many servers because of organizational reasons.

4.2.3.2 TCP forwarding

+ A very low resources are needed by the firewall server to perform the pure forwarding of TCP connections.

+ The application can take advantage of the abilities of the client workstation on which it is running.

+ Using the compilation library, in order to integrate the Internet service access functionality into Unix network applications, results in a low modification effort, while, using the runtime library results in none.

+ Just one TCP/IP module is required on the client workstation, where, an export of file systems via NFS is not necessary which makes it possible to support systems that are not running under Unix.

- A restriction of services is nearly impossible, since, the user can choose freely among network service programs.

- The Internet service access on the transport layer is restricted to the transport protocol TCP.

4.3 SURF Firewall

Stanford University Research Firewall (SURF) [31] , developed at the European Institute for System Security (E.I.S.S.), evolved over a long period of time, more than two years. Because, trade-off between safety and collaboration in a research environment is unacceptable, the SURF firewall is designed with the objective of balancing the collaboration and security concerns for academic and research environments. Therefore, while maintaining information security, the free exchange of ideas is still supported.

4.3.1 The SURF Security Policy

Every firewall has its own security policy that trades off between the need to support collaborative work and the need to provide security. This is because collaborative work requires open Internet connections while providing security requires restricting this connectivity. The security policy is limited by implementation considerations and is affected by how much trust is given to internal users. The SURF firewall policy attempts to balance

the collaboration and security concerns better than in traditional corporate firewalls. The security policy can be stated in three simple rules as shown in Figure 4.5:

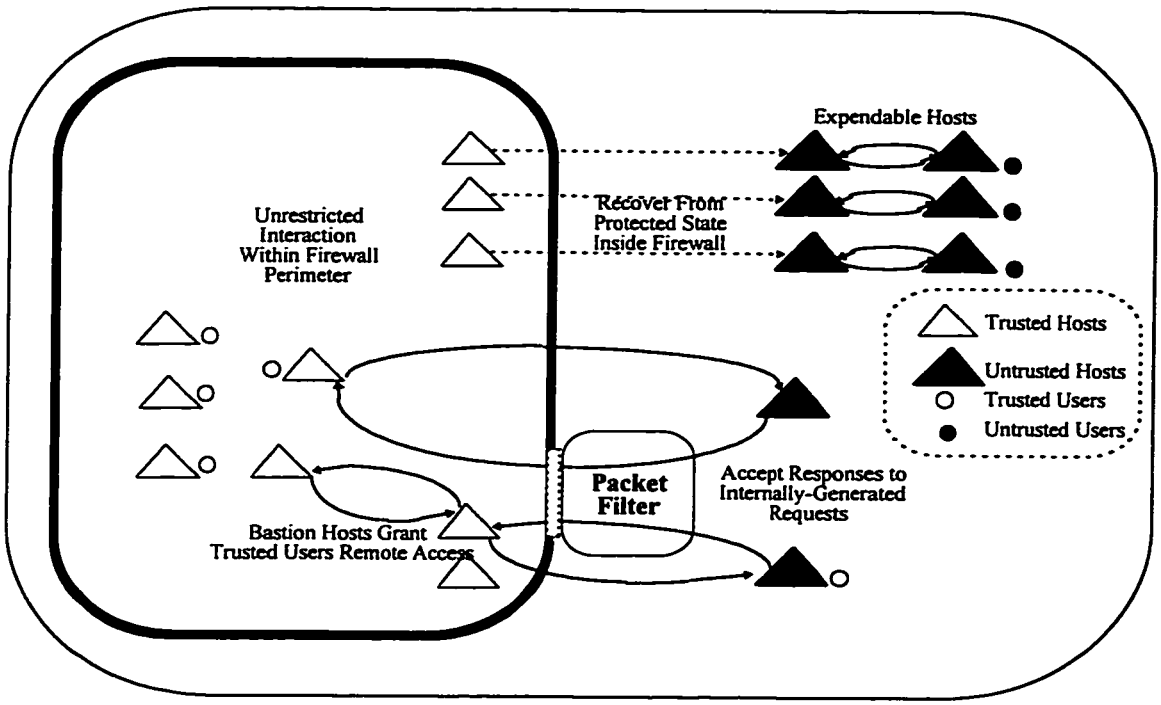


Figure 4.5. SURF Design with a Request-Response Security Policy, Expendable Hosts, and Bastion Hosts Supporting Remote Access for Trusted Users.

- **Request-Response:** All outbound packets are allowed to travel outside. Inbound packets are allowed inside the firewall only if they can be determined to be responses to outbound requests.

- **Expendable hosts:** Because they are outside the security perimeter, packets to or from outside-the-firewall are unrestricted (aside from normal operating system and application-level access controls).

- **Virtual Enclave:** Packets known to be from authenticated hosts or users outside the firewall are allowed inside the firewall.

4.3.2 Request-Response Policy

The implementation of the request-response policy used by SURF relies primarily on a packet filter located at the firewall perimeter. But, some of the responsibilities must be delegated to operating system mechanisms provided by protected hosts to recognize responses to outstanding requests. However, it is relied on application level proxies running on bastion hosts in case when application protocols are not suited for the request-response paradigm.

An incoming packet is detected by the packet filter in order to see whether it is actually a response to a prior outbound request. Packet-filtering hardware is usually stateless, therefore, it decides whether to drop or pass a packet based only on the physical port on which the packet arrives, the port on which it will be forwarded, and data within the packet itself. Packet filtering and host filtering fall into three categories:

TCP Request-Response: TCP-based services that establish explicit connections are best handled by packet filtering, where, connection establishment is treated as a "request" and data transmission on an established TCP connection as a "response". Therefore, external hosts are allowed to participate in TCP connections established by any host inside

the firewall. While, external hosts may be prevented from initiating such connections, by filtering incoming connection requests and dropping extraneous TCP segments by internal hosts.

Connection-Based UDP Services: UDP-based application protocols that use well-known kernel ports at both the client and server hosts are treated much like TCP connections. This is achieved by deploying a heuristic filter based on source and destination addresses, port numbers, and packet contents and trusting internal hosts to further filter inappropriate responses.

NFS-client filtering at host: NFS is a request-response protocol, since the NFS client sends RPC requests to the NFS server, and the server returns the results. Under the request-response rule of the SURF firewall security policy, NFS clients that are protected by the firewall can perform **NFS-mount** operation to mount file systems from NFS-servers outside the firewall. The SURF firewall supports a request-response policy for NFS, as with TCP, by using a coarse-grain filter and trusting internal hosts to safely ignore and drop unexpected response packets.

4.3.2.1 Application-Level Proxies for Other Inappropriate Application Protocols

Many application protocols are not directly agreeable to the request-response paradigm. These applications typically fall into two categories: connectionless UDP-based applications

and reverse-channel TCP-based applications. Application-level proxies are introduced to create a request-response illusion over an ill-behaved protocol. The proxies run on an internal bastion host configured specifically for this purpose. The SURF firewall permits multiple bastion hosts, thereby, eliminating the performance bottlenecks and allowing each bastion host to be stripped-down to support a limited set of applications. The packet filter accepts incoming packets to these restricted protocols if they are destined to the appropriate bastion host.

Connectionless UDP-Based Applications: Most applications do not use UDP sockets in connection mode, therefore, all packets addressed to the UDP socket of such an application are received regardless of the foreign address or port. Because there is no connection setup, a UDP "response" from an external host cannot be distinguished from a subsequent "request" from an external host that uses the same remote and local port numbers as an earlier response. The packet filter or host operating system should retain some application specific state, in order to evaluate whether an inbound packet is a valid UDP "response". Among the essential connectionless UDP services are NTP for time synchronization and DNS for mapping host names to addresses.

NTP servers are configured in a tree with the higher-level servers periodically broadcasting their current time statistics to the lower-level servers in order for the lower-level servers to adjust their clocks. This server-initiated traffic does not match the request-response paradigm. NTP is allowed through the firewall perimeter by designating a set of

internal hosts as NTP "bastions". The packet filters allow traffic from the NTP port of any external host to that of any NTP bastion. All other internal machines synchronize to the bastion machines.

DNS fits the request-response paradigm and can be configured to use either TCP or UDP. But DNS clients typically use UDP from an arbitrary UDP port, therefore, DNS is managed like NTP. A set of internal "slave" name-servers are configured to forward all requests they receive to a set of external name-servers. The packet filters admit UDP traffic from external hosts with source and destination ports set to the DNS reserved port to the internal name-servers.

For other UDP services, users are simply required to invoke them from the expendable machines outside the firewall.

Reverse-channel TCP-Based Applications: TCP-based services that require a "back-channel" connection, cannot be handled by the request-response implementation. An example of this is the FTP, because, FTP clients implement requests like **dir** and **get** by:

1. Creating a local "throwaway" TCP socket,
2. Sending a message to the remote server asking that the resulting output be sent to the throwaway port,
3. Waiting for the remote server to connect to the throwaway port and send the data.

The back-channel problem arises in a different form when a user inside the firewall tries to start an X windows client on an external host to appear on the user's local display. To connect to the display, the external client must establish a TCP connection to the X server inside the firewall. The SURF firewall access policy does not allow this connection.

The use of application-level proxies is regarded as a temporary workaround for ill-designed application protocols.

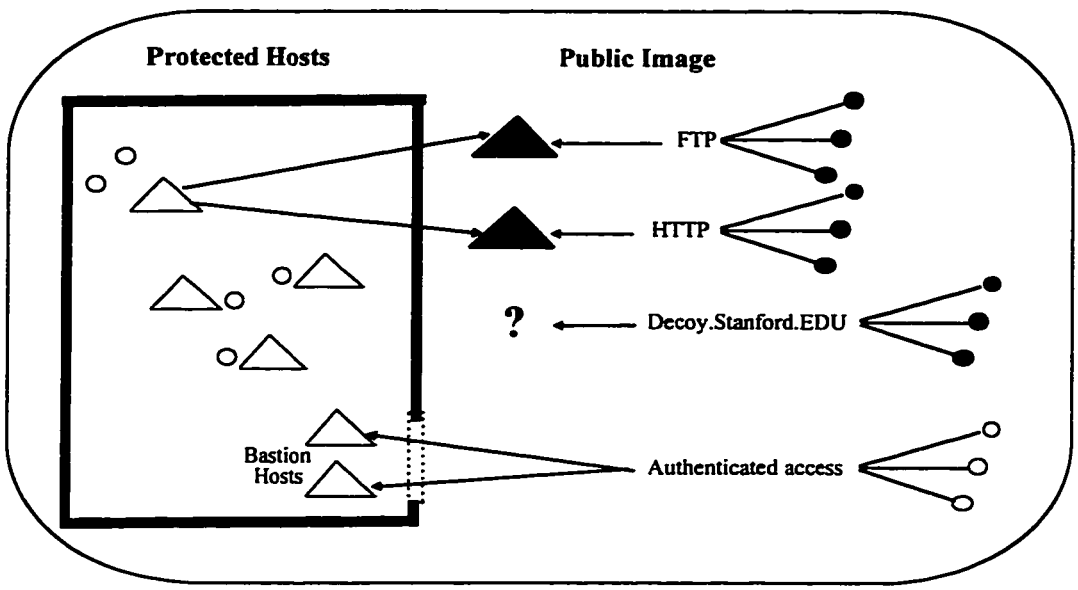


Figure 4.6. Only Expendable Hosts and Decoys are Exposed to the Internet

4.3.3 Exposing a Secure Public Image

The public image consists only of "expendable" hosts and decoy host names, as shown in Figure 4.6. An expendable host is a machine outside the firewall, whose data can be easily re-created from information kept securely behind the firewall. On the other hand, a

decoy host name points either to a non-existent machine or to a machine that simply log all accesses. DNS entries for any of the machines located inside the firewall are not published except for bastion hosts. The public image is designed to support untrusted interactions with the wider Internet community. These interactions fall into three categories:

1. Information Publication
2. Insecure Software and experimental protocols
3. Supporting Guest Users

4.3.4 Secure Non-Local Access

The SURF firewall provides two methods of non-local access with different levels of functionality:

1. **Secure IP Tunneling** which provides authenticated secure IP access. It also allows the remote user to freely send IP packets without restriction through the firewall.
2. **Remote Login and Remote Execution** that provides authenticated remote login.

intended for users who are traveling or working off-site but cannot set up a secure IP channel.

4.3.5 Vulnerabilities of the SURF Firewall

There are three potential sources of vulnerability in the design of the SURF firewall. These are (1) the open environment behind the firewall and absence of outgoing packet filtering, (2) the coarseness of the incoming packet filter, and (3) the potential for hijacking acceptable connections.

4.3.5.1 Open Research Environment

The first risk arises because the traffic between machines behind the firewall is not restricted and because no outbound packets are filtered. As a result, if an intruder penetrates the security perimeter, he will have unrestricted access to all internal hosts. The SURF firewall does not prevent any outbound operation, including data transfer. Therefore, once an intruder gains access to an internal host, the firewall no longer offers any protection. For instance, if an intruder gains access to an internal host, he may download, into the protected network, an executable that performs some forbidden operation. However, any attempt to limit these risks would significantly affect the openness of the research environment.

4.3.5.2 Coarse-Grain Packet Filter

The second risk comes from the use of a coarse-grain packet filtering while relying on internal hosts to perform additional filtering. As a result, the packet filter admits packets that may not be responses to outstanding requests. This may expose the internal network to

a denial-of-service attack flooding the network with extraneous packets that internal hosts must process and drop. However, this risk is not significant, since an intruder can flood any protected network by discovering a packet which obtains a response from behind the firewall.

4.3.5.3 Connection Attacks

The last potential risk arises from attacks to authorized connections through the firewall, such as TCP sessions to external hosts. The SURF firewall is vulnerable to this kind of attack because it supports authenticated access through potentially insecure hosts and networks outside the firewall. For an example, NFS requests to a compromised external file server may cause the firewall to read or execute intruder-supplied data. To address these dangers, care should be taken in selecting applications to be made available to users. For example, applications with response packets that could obtain control of a shell running inside the firewall, should not be permitted through the firewall.

4.4 TIS Internet Firewall Toolkit (FWTK)

The TIS Internet Firewall Tool-kit [5] consists of software modules and configuration guidelines developed in part during the course of a project sponsored by the U. S. Department of Defense, Advanced Research Projects Agency (ARPA). The Tool-kit components are designed to work together, but can be used in isolation or be combined with other firewall

components as well. The software of the Firewall Tool-kit runs on UNIX systems using TCP/IP with the Berkeley socket interface.

For a wide range of services, such as X, FTP, TELNET, there exist some kind of proxies (see Figure 4.2). The most security benefit of using proxies is that they require authentication. The proxy protects the firewall host itself, because it does not allow the user to login to the firewall, and by permitting only authenticated users to gain access from the outside, it protects the network. Other store-and-forward services, such as Internet (SMTP) mail and USENET news, fit the proxy approach to firewalls. Sometimes, such service daemons run with system privileges and may contain bugs that an attacker can exploit. Also, the server itself may in some cases cause a compromise of the network security. In the design of the TIS Internet Firewall Tool-kit, it is attempted to avoid these problems by:

- using proxies that can be locked into a specific sub-directory while running by means of **chroot**[13].
- designing proxies to run without special system privileges, in order to further reduce the chance of causing damage to the system.

An outside user, ideally speaking, should not be able to interact with a privileged process. However, in practice the Internet service master daemon **inetd**[14] needs to run with privileges, but outside users cannot interact directly with it.

[13] **chroot** is a UNIX system call that permanently restricts the working file-system of a process.
[14] **inetd** daemon is responsible for starting other service daemons.

4.4.1 Design Philosophy

Many existing firewall systems are based on "known to be good" software or that is considered to be trustworthy, because it has been used extensively for a long time. But, the "known to be good" approach has not been very reliable, since no matter how carefully they are maintained, certain software components are frequently exploited by intruders. In addition, these programs are usually complex software and require system privileges in order to operate.

As a step towards addressing this problem, the TIS Firewall Tool-kit is designed to be verified for correctness as a whole or at the components level as well. Its components are implemented as simple as possible. Since the Tool-kit components provide a single service each, they may be examined separately. The following general firewall design principles are considered in operating the firewall Tool-kit:

1. Services that are misconfigured should not work at all. In this way, a network service will not compromise the system if there is a bug in its implementation.
2. Network services that are running with privileges should not be directly connected to by hosts on the untrusted network.
3. Implementation of network services is made with minimum features and complexity. Therefore, the source code can be reviewed thoroughly and quickly because of its simplicity.

4. Testing the system for correct installation should be accomplished with reasonable and practical means.

The Tool-kit is designed to be used with a host-based security policy, it is possible to use some of its components with router-based firewalls. In router-based firewalls, routers can be configured to provide additional paths between the protected network and the Internet as needed. In this manner routers permit the Tool-kit software, running on a secure host, some degree of access between the Internet and the protected network. The additional paths are outside the control of the Tool-kit and should be carefully studied.

In this section, we will focus on the host-based approach. The security of the host, in a host-based firewall, is crucial and once it is compromised the entire network is then vulnerable to attacks. But, because of the ease with which it can be maintained, configured, customized and audited, a host-based firewall is considered to be superior to other solutions.

For extra security, the Tool-kit may be used in conjunction with router-based screening. In order to minimize risks, the provided services on the external machine (bastion host) are sharply reduced and reviewed. Other proxies such as Digital Equipment Corporation's X Window System proxy [47] can be added to the architecture of the standard firewall configuration on which the only supported services may be as follows:

- SMTP service is supported through a non-privileged front end that runs locked in a "safe directory" via chroot.

- FTP is supported via a proxy that runs without requiring special privileges.
- NNTP is supported via a "tunnel" server that permits traffic between a host on the inside and its news server on the outside.
- TELNET service is via a proxy that runs unprivileged.

In this case, only these four services must be analyzed for risk, since all other services on the system can be disabled selectively. Trust in the system can be increased by running all services unprivileged and analyzing the security of each service in isolation. The overall security is irrelevant to the security of individual services since they are running in jailed modes.

4.4.2 Configuration and Components

The tool-kit can be installed as shown in Figure 4.11, as an environment that combines a bastion host firewall with routers. In this environment, the firewall and the routers share the implementation of the security controls. While, the routers are controlling network-level access, the bastion host performs the control at the application-level. Another, but simpler firewall configuration may consist of only a dual-homed gateway, where, a workstation with two network interfaces and disabled IP forwarding is connected to both networks. Comparing both implementations we can see that:

- The dual homed gateways are less flexible than firewalls that have routers and hosts

combination. This is because, the option of routing services at the network level is generally not available.

- Since routers are not integrated in the firewall Tool-kit security system, a dual-homed gateway has a higher degree of confidence that no network traffic will leak through a router.

The Tool-kit has the following design considerations:

- A host-based firewall is built with security enforced by a single bastion host.
- All the proxies and access control tools are made via a single configuration file with a regular syntax.
- The configuration rules (Figure 4.7) in top-to-bottom and left-to-right precedence provide information for both configuration and service access permissions.
- In the configuration rules, host-names or IP addresses including simple wildcards can be used, but since DNS addresses are vulnerable to spoofing, IP addresses are preferred.

Since direct traffic is not permitted between outside and inside systems, services that use the connectionless point-to-point UDP protocol are not allowed and therefore cannot be used through network proxies. However, UDP-based services such as NTP and DNS can be provided transparently through the firewall by configuring UDP servers to forward queries that originate within the protected network

```
#Example ftp gateway rules:
#-------------------------

ftp-gw: authserver 127.0.0.1 7777

ftp-gw: denial-msg /usr/local/etc/ftp-deny.txt

ftp-gw: welcome-msg /usr/local/etc/ftp-welcome.txt

ftp-gw: help-msg /usr/local/etc/ftp-help.txt

ftp-gw: timeout 3600

ftp-gw: permit-hosts 192.33.112.100

ftp-gw: deny-hosts 128.52.46.*

ftp-gw: permit-hosts 192.33.112.* -log { retr stor } -auth { stor }

ftp-gw: permit-hosts * -authall
```

Figure 4.7. Example ftp gateway rules

4.4.3 TCP access control

With respect to TCP access and use, most network processes on BSD-based UNIX systems are started by an initial connection to inetd which acts as a general-purpose network listener. When contacted, inetd establishes a connection between the incoming request and the program which will service the request. For example, when an incoming request for the TELNET service is heard by the running network listener, a program is executed and connected to the incoming request. This program is selected according to the entry for TELNET in the inetd's configuration file. The only function that the Internet services daemon (Inetd) performs is to invoke specified processes to manage the requested network services. Some vendor implementations permit system administrators to specify a user-id

to be used when invoking the desired service, but there is no provision for access control based on the request point of origin.

On the Internet, a variety of implementations of "wrapper" processes are available with varying functionality. The "wrapper" process used by the Tool-kit is called "netacl" where it provides support for all TCP-based services. With only TCP-based services supported and UDP services disabled so that they would be no longer a worth worrying threat, Netacl does not have great advantages over other versions of TCP wrappers. The advantages that the Netacl has over other versions of TCP wrappers are:

1. Its minimal size (240 lines of code, including comments and a large copyright header),
2. Its intentional lack of UDP support,
3. Its sharing of a common configuration mechanism with the other tools in the Tool-kit.

4.4.3.1 TCP Plug-Board Connection Server

In the situations where services are provided through a firewall, the administrator has the choice to either run the service on the firewall machine itself or install a proxy server. An example of this is the Usenet news service which, if run on the firewall itself, might expose the system to any bugs in the news software. Therefore it is safer to use a proxy to gateway the service onto a "safe" system on the private network.

Plug-gw is a configurable general purpose proxy that transparently "plugs" two services together. It only acts as a data pipe, therefore, it performs no local disk I/O and invokes no sub-shells or processes. The primary use of the plug-gw proxy is to support Usenet news, however, it can be used as a general-purpose proxy. It also logs all connections, just like other proxy servers do. Caution should be undertaken when plug-boarding TCP connections through the firewall because plug-gw uses only the host address of the client for authentication and no other means and does no examination of the traffic passing across it.

In the case of NNTP, for example, a security defect in the NNTP server on the internal host can be exploited if no firewall is used to make it much harder for an attacker to gain access to the internal system to further exploit the hole. However, if the flawed NNTP server were running on the firewall bastion host itself, the entire firewall might be vulnerable. It has been found from system administration standpoint that news administration is simplified by running it as readily accessible internal server. Alternate approaches, such as modifying the news server to run "chrooted" are potential areas for future research.

4.4.3.2 User Authentication

A generic authentication service is provided by the network authentication server **authsrv** for all proxies of the Tool-kit. It is used optionally except when the firewall FTP and TELNET proxies are configured to require authentication, then its use is required.

Authsrv integrates multiple forms of authentication, where, it permits administrators to associate a preferred form of authentication with individual users. This way organizations are permitted to enable the same token of user authentication to the firewall if they already provide users with authentication tokens. Authsrv also provides a simple programming interface to the authentication service, because most commercial authentication systems have unique nonstandard interfaces. It supports several forms of challenge/response cards along with software-based one-time and plaintext password systems. Due to the threat of password sniffing attackers, however, use of plaintext passwords over the Internet is strongly discouraged.

A simple administrative shell is included to permit manipulation of the authentication database over the network. Authentication transactions can be optionally encrypted. a basic form of group management is supported by the authsrv database, so that one or more users can be identified as the administrators of a group of users, where they can add, delete, enable, or disable users within that group. Authsrv is intended to run on a secured host, such as the bastion host itself, since its database must be protected from attacks. The database has the following features:

1. It internally maintains information related to the last time a user was authenticated and how many failed attempts have been made.
2. It can automatically disable or time-lock accounts with multiple failures.

3. It maintains extensive logs of all authsrv transactions.

4.5 SOCKS Firewall

The SOCKS package is an Internet socket service. It consists of three parts: (1) client library routines, (2) a daemon, and (3) a simple protocol to provide a suitable and secure network connectivity through a firewall host. Although, there are several possibilities for setting up a firewall, the SOCKS package provides a convenient, simple, and vendor-independent solution that maintains the integrity of the firewall. However, the SOCKS package does not enhance the security of the host it runs on. These features make SOCKS a suitable mechanism for securing Internet accessibility through a firewall and providing a more secure access method to the local network [42] .

Modification of client software applications can be made easily to replace the normal socket library calls with the SOCKS library routines. By doing so, all outgoing connections will go through the SOCKS daemon (sockd) which runs on the firewall host. When SOCKS package is run on the firewall host, users will not notice any apparent difference between the use of SOCKS and the use of regular client software. Where all connections at the application level work in the same way with or without SOCKS except for the transparent passing of all traffic through the sockd on the firewall host. Therefore and because of applications ability to replace the normal socket library calls with the SOCKS library

routines, a transient and transparent firewall process can be automated and utilized as a single point of access to the Internet [42] .

4.5.1 SOCKS Library

With SOCKS library calls, all connections are made via the sockd daemon on the firewall where information is transmitted and all network operations are performed by the daemon. The daemon passes any data it receives from external connections to the internal requesting host. Therefore, to the internal host, all communications appear to be normal, while they appear to the external host as if they are originating from the daemon [42] .

In the SOCKS library, the normal C library socket calls are replaced by the same names but prefixed with an "R". Also the same parameters are used as in the original functions except for the Rbind function where the address of the remote host from which the connection will be established is an additional parameter. This additional parameter is needed so that the daemon can refuse unauthorized connections. The complete list of SOCKS functions are: Rconnect, Rbind, Rlisten, Rgetsockname, and Raccept. All of these functions are designed to propagate all network connections to the SOCKS daemon running on the firewall [42] .

4.5.2 SOCKS Protocol

The SOCKS protocol is used between the daemon on the firewall and the SOCKS library routines on the internal hosts. It consists of only two commands, the first one is the CONNECT command which requests the daemon to establish an outbound connection to a given address and port number. While the second command is BIND which requests an expected inbound connection from a given external address. For purposes of logging, the requester's user-name is passed from the requesting host to sockd. The syntax of the two commands are shown below [42] :

CONNECT <IP-Address> <Port-Number> <User-Name>

BIND <IP-Address> <User-Name>

4.5.3 SOCKS Daemon

The Internet service master daemon (inetd) is used to start the SOCKS daemon (sockd) on the firewall host to accept connections from approved hosts only as determined by the configuration file. The SOCKS library routines may be used by the applications running on these hosts to communicate with the SOCKS daemon. The daemon operates as a transient point for socket connections and carry out the requested actions through the SOCKS protocol commands CONNECT and BIND. It logs the originating host and the user name for all connection establishment attempts. Figure 4.8 shows a typical example of how a write() to a SOCKS socket would be performed [42] .

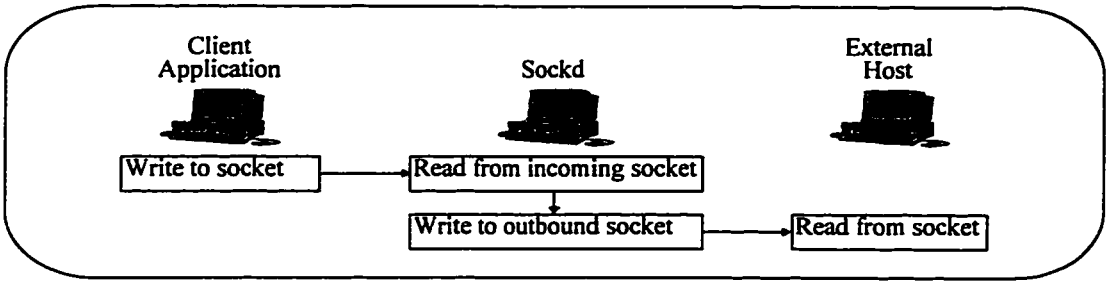


Figure 4.8. Sockd as a transient socket server (relative event timings are indicated by the row position of the event text box)

Figure 4.9 shows how the CONNECT request works. The Rconnect() routine is called on the internal host whenever the client application wants to generate a CONNECT request. Which in turn causes the daemon to establish the desired connection to the remote host. The daemon then returns a connection status of success or failure. After the establishment of the connection, the sockd will act as a bridge between the local and external socket connections, where, the application reads and writes to the local socket connection of the firewall [42] .

BIND requests follow the same fundamental idea, but is a little bit more complex. As shown in Figure 4.10, the sequence of events that are followed by the BIND request are as follows:

- The client application calling the Rbind() routine to connect to sockd.
- The sockd daemon validates the connecting host and binds a new socket connection to a free port on the firewall.
- If the connection establishment is successful, the sockd returns the firewall port to the

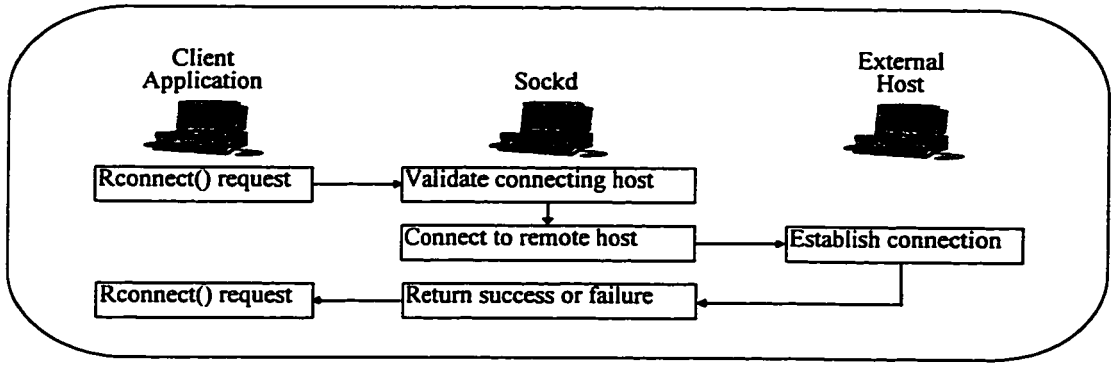


Figure 4.9. Socks CONNECT Request (relative event timings are indicated by the row position of the event text box)

client application for which this connection was bound.

- The daemon then performs listen() and accept() and assumes that the client application will execute a bind command followed by these actions:

  * The client application can then call RIisten(),

  * Followed by a call to Raccept() which waits for the daemon to transmit a second packet. This second packet contains the remote host address and port from which a connection was established, as well as, any failure that might be caused by either a resource failure or a connection received from a different host than specified in the BIND request.

- Having completed these steps, all reads and writes to the socket can pass through the firewall between the internal and remote hosts [42] .

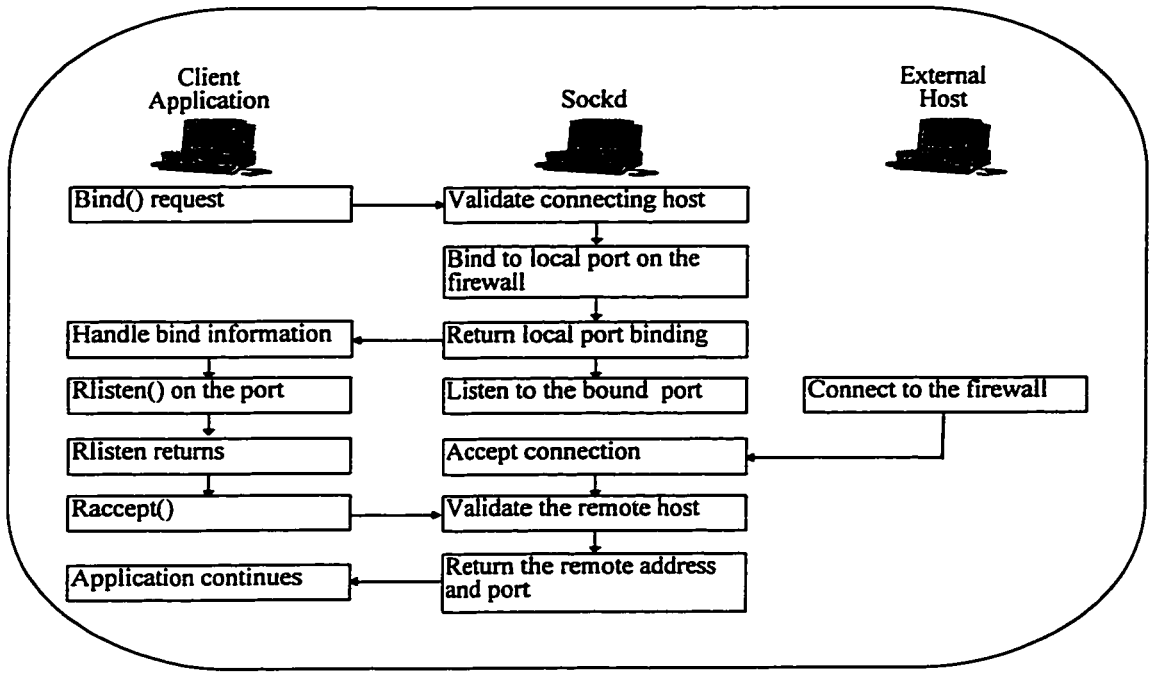


Figure 4.10. Socks BIND Request (relative event timings are indicated by the row position of the event text box)

4.6 Summary

In this chapter the following examples of firewall products were presented: (1) Digital's Three Way Isolation, (2) IpAccess Firewall, (3) SURF Firewall, (4) TIS Internet Firewall Toolkit (FWTK), and (5) SOCKS Firewall

CHAPTER 5

FIREWALL EXPERIMENTAL EVALUATION AND COMPARISON

Firewalls provide a sense of security and can be critical in enforcing the security policy that governs what should traverse the Internet link. In an attempt to benefit from the situation arising from user's fear that intruders may break into their corporate network through the Internet, several vendors have released dozens of firewall products.

5.1 Previous work

Some firewall products offer more features than just packet filtering with varying capabilities, therefore, firewall evaluation and comparison has been discussed by a number of computer related articles [14, 28] .

The result of these comparisons showed that some firewall products offer more features than some others. For instance, using the virtual private network capabilities of the FireWall-1, CyberGuard, and Network Systems Security Router products, communications between the designated sites will always be encrypted. A less secure level of encryption is offered by the KarlBridge product which is probably adequate to prohibit casual snooping. Another important feature that is offered by FireWall-1 and CyberGuard is the network address translation. In addition, all products offered some reporting and logging features, which

can help network administrators log attacks and, in some cases, notify them when attacks are occurring. On the other hand, logs may be used simply to find out what types of sites and services are taking most of the time on the Internet link [14] .

In this section, we will examine different firewall products using: (1) some security evaluation tools in order to automatically explore their weaknesses and strengths and (2) observations based on interaction with the firewalls being evaluated. We will also, establish a testing methodology to facilitate such work in the future. In addition to the above an attempt will be made to investigate the customizing flexibility of the examined firewalls to meet some anticipated academic and corporate requirements.

We will choose two firewall products for evaluation using one security evaluation tool. The two firewall products will be compared based on some standard features such as performance and filtering power. The objective of this work is to point out the strengths and weaknesses of the examined firewalls and to establish a testing methodology to evaluate other firewall products that are available in the market.

5.2 Test-Beds

The only test-bed that we found in literature is shown in [28] . This test-bed is designed for performance evaluation of individual firewalls one at a time.

In this section, we propose three different test-beds for the evaluation and comparisons of firewalls. Also, we will show the advantages and disadvantages of each of these three test-

beds. These test-beds are used to provide a backbone architecture to facilitate connectivity between host(s) running the examined firewall(s) and the host(s) running network analysis or hacking simulation tools.

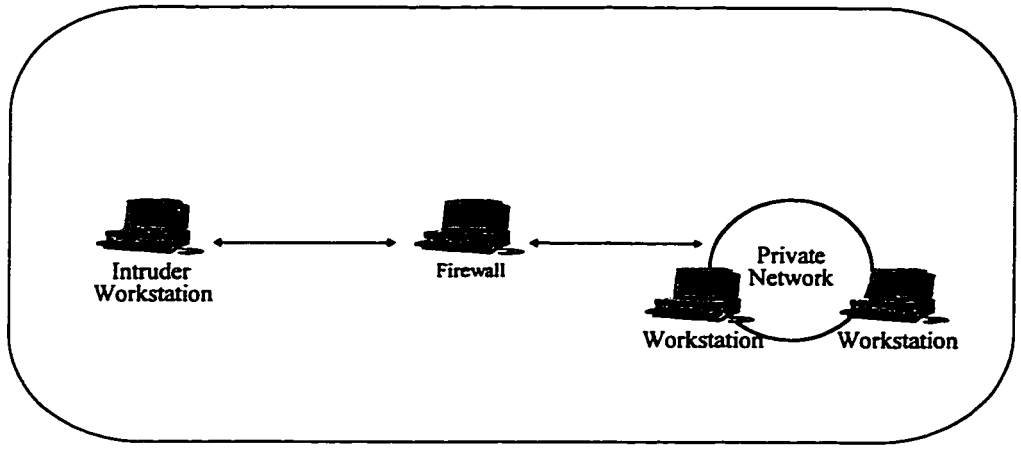


Figure 5.1. Test-Bed 1

- The first and simplest test-bed consists of an intruder workstation, a firewall workstation, and a private network as shown in Figure 5.1. The security analysis tools are run on the intruder workstation to attack the private network which is protected by the firewall. In this case one firewall can be tested at any given time.

- The second test-bed consists of a router connecting two private networks, where each network is protected with one of the two firewalls under test, as shown in Figure 5.2. In this case, the intrusion software is run from the inside of each network to attack the other.

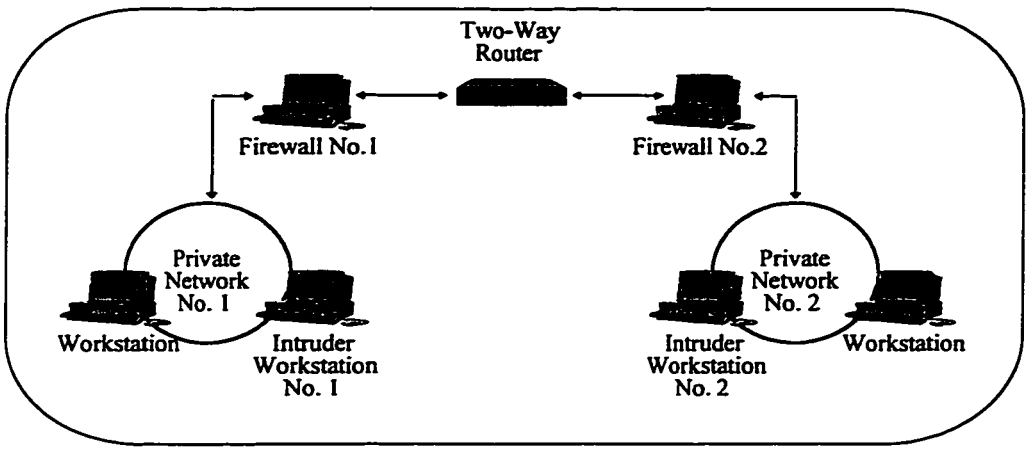


Figure 5.2. Test-Bed 2

- Finally, the third test-bed consists of two private networks and one untrusted network, all connected through a three-way router, as shown in Figure 5.3. One of the two firewalls under evaluation is used to protect each of the two private networks. The intrusion software is run from the untrusted network to attack the other two networks.

5.2.1 Comparison of Test-Beds

Comparing the first test-bed to the other two test-beds, we can see that it is:

1. The cheapest in terms of the required hardware.
2. The most expensive in terms of the required test time.
3. The most inconvenient solution, because the system must be reconfigured and restarted whenever the tester decides to change the firewall under test.

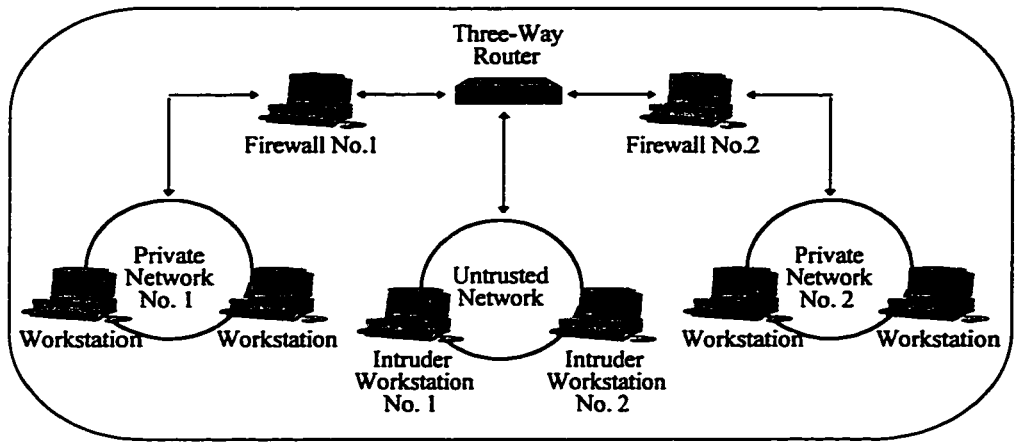


Figure 5.3. Test-Bed 3

The second test-bed requires more hardware compared to the first test-bed. On the other hand, it is faster than the first test-bed because the two firewalls are tested simultaneously. However, the testing conditions are not expected to be completely identical, since it is not guaranteed to have 100% compatible networks, specially, when considering the effect of the installed firewalls on the outbound traffic generated by the analysis tool running behind it. For instance, running SATAN behind a firewall requires unsetting some environment variables or changing the browser's configuration so that it will not use the SOCKS or HTTP proxy host [27] .

The architecture used in designing the third test-bed is the most expensive compared to the other test-beds, however, in addition to the features found in the second test-bed, the test is performed under similar conditions and more than one security analysis tool can be used simultaneously to attack the private networks.

Although, the three proposed test-beds were designed to test two firewalls, they can be used to test any number of firewalls. Test-bed 1 can be used to test any number of firewalls, but only one at a time. The other two test-beds can be easily modified to achieve concurrent multi-firewall testing, simply by adding one branch for each additional firewall to be tested.

5.3 Selection Criteria

After reviewing the literature of some firewalls as shown in chapter 4, we decided to choose SOCKS v5 and TIS FWTK, version 2.0, as the two firewalls to be evaluated and examined. Reasons behind our decision for this selection are:

1. They are commonly used,
2. They are public domain,
3. We have their source code.

We have decided to use SATAN as our security analysis tool, for the same reasoning used in justifying the selection of the two examined firewalls.

The architecture used in the design of the third test-bed makes it the most convenient test-bed. More importantly, it provides similar test conditions for both firewalls being tested. For these reasons, we have chosen the third test-bed to test the two selected firewalls.

5.4 Setup of The Selected Test-Bed

The objective of the test is to attack two firewalls using some security analysis tools in order to compare their strengths and weaknesses. Therefore, the selected test-bed is used with only one workstation to host the security analysis tool on the external network.

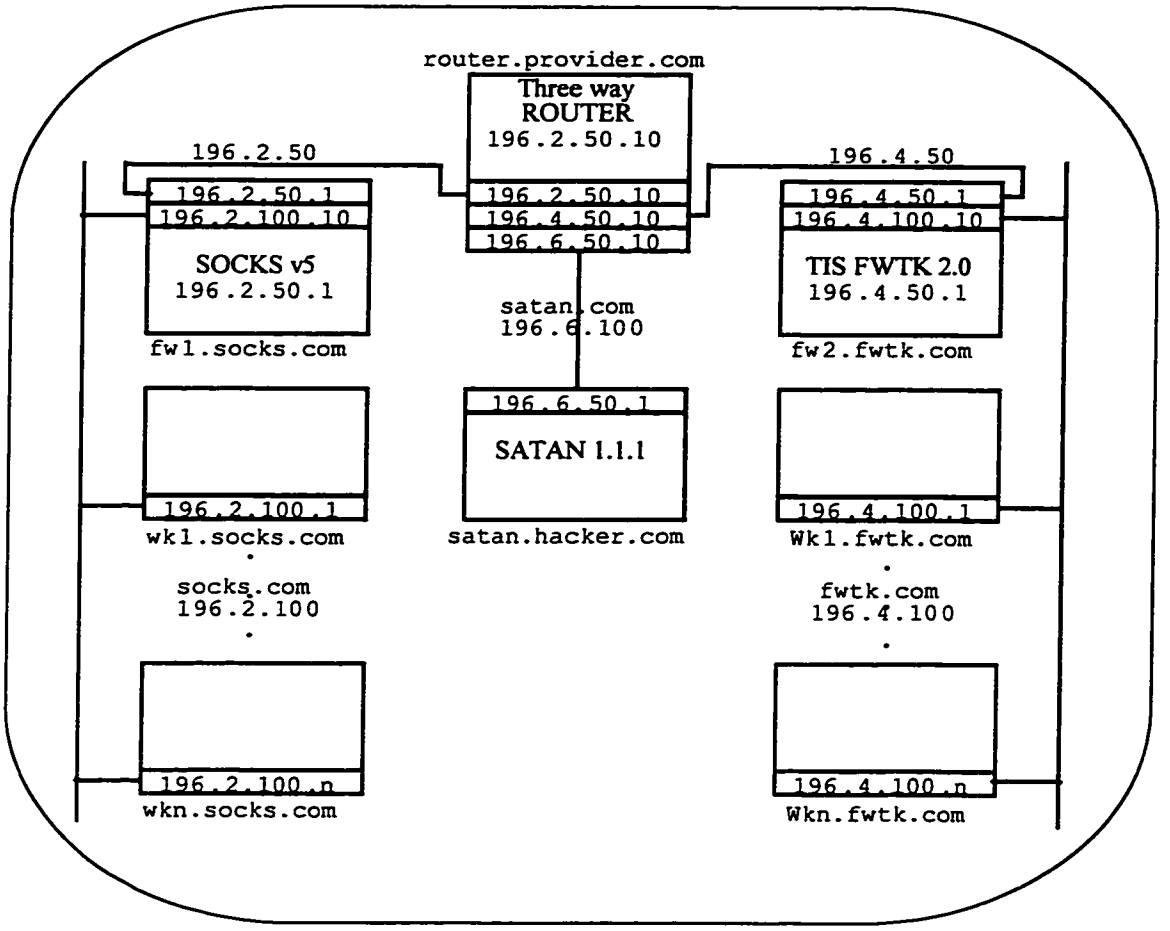


Figure 5.4. Setup of the Selected test-bed

A number of network setup and configuration files are used to setup the selected test-bed as it appears in Figure 5.4. In addition, some output files are generated to ensure the

successful compilation and installation of the selected firewalls and security analysis tool. The architecture of the test-bed is as follows:

- A three way router "router.provider.com" with IP address 196.2.50.10. The IP addresses of the three Network Interface Cards (NICs) installed on the router are 196.2.50.10, 196.4.50.10, and 196.6.50.10. They are connected to the SOCKS firewall host, TIS FWTK host, and SATAN host respectively.

- A dual homed host "fw1.socks.com" with IP address 196.2.50.1. The IP addresses of the two NICs that it has are 196.2.50.1 and 196.2.100.10. The first one is connected to the three way router, while, the second one is connected to the internal network "socks.com". The SOCKS v5 firewall is installed and run on this host.

- A second dual homed host "fw2.fwtk.com" with IP address 196.4.50.1. The IP addresses of the two NICs that it has are 196.4.50.1 and 196.4.100.10. The first one is connected to the three way router, while, the second one is connected to the internal network "fwtk.com". The TIS FWTK 2.0 firewall is installed and run on this host.

- The "satan.hacker.com" with IP address 196.4.50.1 is the only host on the external network "hacker.com". The SATAN host is connected to the three way router. The SATAN 1.1.1 analysis tool is installed and run on this host.

5.5 Test Methodology

Our test methodology is simple. It consists of two parts: (1) an automated test using a security analysis tool on the external network to scan firewall hosts and protected networks for any vulnerabilities, and (2) making observations based on manual interaction with the two firewalls from inside and outside in all possible ways.

5.5.1 Test Conditions

In order to facilitate comparative analysis, the results of the automated test of the two firewalls and their protected networks are generated under three different test conditions as follows:

1. **Enabled IP Forwarding:** In this case the IP forwarding is enabled on the dual homed host. No firewall protection is needed because it will be bypassed, any way, by the forwarding mechanism at the network layer.

2. **Disabled IP Forwarding:** In this case, the IP forwarding is disabled on the dual-homed host and no firewall protection is provided. The objective is to see if SATAN can reach any workstation behind the dual-homed host in the absence of firewall protection.

3. **Firewall Protection:** Finally, we will use firewall protection while disabling IP forwarding and see what the security analysis tool will report about the vulnerabilities on firewall hosts and the protected networks.

To facilitate switching between these different test conditions, we have saved two copies of LINUX kernels (one with enabled IP forwarding and one with disabled IP forwarding). In addition we have created some scripts to automate the switching process. And in order to make things more easier and transparent we have also defined some aliases. These useful scripts and aliases are shown in appendix "B".

5.6 Test Results

When comparing TIS vs. SOCKS, analysts would like to know what are the pros and cons of each. As mentioned earlier in section 5.5, our testing methodology consisted of automated and manual approaches. The automated results are based on using SATAN security analysis tool. While the second part is based on interactive observations.

5.6.1 SATAN Based Results

SATAN has a list of vulnerabilities that it checks for (see section 2.4.1). When one of these vulnerabilities is discovered on one of the machines that SATAN is able to reach, it will be reported along with identity of the visited workstations and suggested solutions to the problem. Otherwise, SATAN will report the identity of the scanned workstation and a message indicating that there are no vulnerabilities found.

In this automated firewall evaluation, SATAN reported no vulnerabilities on any workstation including the two firewall hosts. It also could not penetrate any of the tested

firewalls in order to reach any protected workstation. Therefore, we can assume that both firewalls can stand attacks similar to those generated by SATAN.

We went one step further to check what SATAN can and can not do. We removed the firewall protection and ran SATAN against the two unprotected networks. At the beginning, we performed the test while IP forwarding was disabled on the dual homed hosts that were previously used to host each of the firewalls. In this case, SATAN could not visit any workstation behind the dual homed hosts. Therefore, it could not see what vulnerabilities existed there. We repeated the same test again but with IP forwarding enabled. In this case, SATAN visited all workstations behind the dual homed hosts, but found no vulnerabilities.

To make sure that SATAN is actually doing what it is supposed to do, we introduced a hole using the "XHOST" on one of the workstations, which happened to be the SATAN workstation itself, where, the hole was found and reported as it should have been.

In summary, we can say that the tested firewalls stands attacks similar to those generated by SATAN (see section 2.4.1). However, since SATAN did not report any vulnerabilities on any of the two tested firewalls, it can be deduced that SATAN does not really help in the evaluation and comparison of firewalls. Therefore, to use the automated test methodology as explained in section 5.5, we need a more powerful security analysis tool that is not only capable of scanning for a wider range of vulnerabilities but is also capable of exploiting them. This is very important to look for vulnerabilities behind the firewall and not only on the firewall itself.

Comparison Factors	SOCKS v 5	TIS FWTK 2.0
Pre-installation Configuration	Automatic with many options	Manual (requires knowledge of the host system)
Building and Installation	Automatic	Automatic (but individual components may have to be treated separately)
Programs	One single program	Separate small components. This makes the source code simple enough to be reviewed quickly and thoroughly
Affected Machines	The server and all internal as well as external client workstations	The server machine only
Supported Architectures	Single and multiple homed hosts	Single and multiple homed hosts

Table 5.6. Comparison of SOCKS and FWTK Installation

There are several security analysis tools other than SATAN such as the PENETRATOR of Convergence Technologies Incorporation. However, we have used SATAN to perform the automated evaluation of the two firewalls, because, at the time when we started this project, SATAN was the most current and widely publicized. But, as of now, most of the security holes that it checks, have been fixed in recent software. New vulnerabilities may have been introduced with new software, therefore, in similar future work, new and more powerful tools need to be investigated. Such tools should be capable of exploiting vulnerabilities on the firewall to reach protected machines for more security analysis.

5.6.2 Observation Based Results

The test and evaluation results that are based on observation and interaction with the two examined firewalls are summarized in the following tables:

Table 5.6 compares the two firewalls with respect to their installation easiness, flexibility and requirements. The SOCKS firewall has a fully automated installation and a pre-installation configuration procedures, while TIS FWTK requires the administrator to have some knowledge of the hosting system. TIS FWTK components may need to be treated separately, because, some may successfully compile while some may fail. Also, while TIS FWTK is installed on the bastion host alone, the SOCKS firewall must be installed on the bastion host and all its clients must have socksified services or the dynamic socksification library. Both firewalls can be implemented using single or multi-homed hosts.

Table 5.7 compares the two firewalls with respect to their requirements and features during execution time. As shown in the table, SOCKS firewall does not improve the security of the host it runs on. For instance, if telnet daemon is running on the firewall host, a user can bypass the SOCKS firewall by using unsocksified telnet client to connect to the firewall machine. If he has a root access to the system he can do what ever he wants there without the SOCKS firewall awareness. The table, also, shows that TIS FWTK unlike SOCKS can not tell if the connection is originating from inside or outside which means that an external host may impersonate an internal host and be accepted by the FWTK as a legitimate host.

Table 5.8 shows how each firewall behaves with respect to user interaction and usage. It also shows what they can and can not do. In addition, other requirements are shown on the side of the user client workstation in order to communicate with the firewall server.

Comparison Factors	SOCKS v 5	TIS FWTK 2.0
Post-installation Configuration	Simple, using special configuration files. One configuration file is used with SOCKS server and another one is used with SOCKS clients	1) Fairly easy with basic configuration. 2) Can become more difficult to get things working properly with more advanced configuration. Where, it becomes necessary to have good understanding of: a) The individual standard services b) The individual FWTK proxy (gateway) services c) Use of the inetd.conf and services files d) Use of the FWTK netperm-table
Host Security	Not improved	Improved
Clients	Must be socksified in one of the following: 1. Static 2. Dynamic	No special processing is required
Authentication options	Any authentication method	Any authentication method
Access Control options	By rules that identify: 1. Host address of the source and destination 2. Service port of the source and destination 3. User name/password 4. Interface Address	By individual services: 1. Source host 2. User name/password 3. Strong Authentication 4. DNS checking
Interface Detection	Detects network interfaces	Does not detect interfaces
Domain Name Services (DNS)	Not checked	Can be checked to protect against IP address spoofing.
TCP-Based Services	Can be socksified and used.	1) FWTK gateway proxies include: ftp-gw, http-gw, plug-gw, rlogin-gw, tn-gw, x-gw 2) Netacl is used to control other TCP-based services.
UDP-Based Services	Not applicable	UDP services are not allowed. Many UDP-based services such as NTP and DNS can be provided transparently by configuring the servers to act as forwarders for queries originating within the protected network.

Table 5.7. Comparison of Configuration and Execution

Comparison Factors	SOCKS v 5	TIS FWTK 2.0
Firewall type	Circuit level	Application level
Starting, Restarting, and Stopping the firewall	Simple using SOCKS executable on the command line as below	1. Edit /etc/inetd.conf to select desired services. 2. Either reboot the computer, or, follow the step procedure: 3. use: "ps -x | grep inetd" to get the process id of the inetd daemon. 4. use: "kill -HUP <the process id of the inetd>"
Clients	Must use socsified applications	Must connect to the FWTK server before they can connect to the other side.
Transparency	Once configured on the client workstation, it can be made %100 transparent to the users	Not transparent, where users must connect to the FWTK server before they can connect to the other side of the firewall
Inbound Connections Accepted From	Other socks servers and clients only	Any TCP/IP workstation including SOCKS servers and clients
Connection through the other firewall	SOCKS clients can connect through the FWTK server	FWTK clients can not connect through a SOCKS server
Firewall Host Accessibility	Does not allow access to the firewall host	Can be allowed or denied in various ways, where there is a tradeoff between convenience and the risk of attacking the firewall host. However with strong authentication, this risk can be minimized.
Allowed Services	Any statically or dynamically socksified application or services	Only FWTK proxy services or standard services running under the control of the FWTK firewall server.

Table 5.8. Comparison of SOCKS and FWTK Usage

PROS	CONS
Transparent	Circuit level: Controls are made only at connection time
Socks clients can connect through other TCP-based non-SOCKS firewall servers.	Connections are accepted only from other SOCKS servers and clients
Authentication and Encryption	Destination SOCKS firewall address must be known to the local SOCKS server and specified in its configuration file
Access control by source/destination host/port combinations	Local SOCKS firewall address must be known to its clients and specified in their respective configuration file.
Interface Detection	Only socksified application or services can connect through the firewall.
Connections can not be made to the firewall host (this is considered as an advantage because the firewall is of a circuit-level type)	Does not enhance the security of the host it runs on.
	Lack of UDP support

Table 5.9. Pros and Cons of SOCKS

Tables (5.9 and 5.10) provide a different angle for looking at the information found in tables (5.6 through 5.8).

Table 5.9 summarizes the pros and cons of the SOCKS firewall. Through the use of statically socksifiied applications or aliases such as (alias telnet=<socks-path>/bin/runsocks talent) to dynamically socksify applications, 100% transparency can be achieved. Socksified application will continue to work on the internal network, even if the SOCKS firewall server is down, therefore, achieving another level of transparency.

Table 5.10 summarizes the pros and cons of the TIS FWTK firewall. The FWTK is the basis for TIS commercial product, Gauntlet.

PROS	CONS
Application level	Not Transparent
Authentication, Strong Authentication, and Encryption	NO UDP Support, however, UDP services can be made available through tunnels using the plug-gw.
Access control by source and destination host/port combinations	No Interface Detection
Connections are acceptable from Any TCP/IP workstation.	
Any application can connect through the firewall if the proper service is provided.	
DNS checking	
Enhances the security of the host it runs on.	
Firewall host can be made available for remote connections. This is considered as an advantage because: 1) FWTK is an application level firewall type 2) Strong authentication can be used	

Table 5.10. Pros and Cons of FWTK

5.7 Summary

In this chapter, an experimental testing and evaluation of SOCKS and TIS FWTK was performed. In preparation for this test, new test-beds were proposed, where, one of them was used and implemented. A new testing methodology was formulated to be used along with the selected test-bed. The results of the test were presented in two parts: (1) automatically generated by the security analysis tool, (2) manually observed based on interaction with the tested firewalls.

CHAPTER 6

PROPOSED FIREWALL SOLUTIONS

In this chapter, we present and discuss two proposed firewall architectural designs to achieve secure connectivity to the Internet. These proposed architectures can be used as the basis for in-house development of firewalls that meet wide range of academic and corporate requirements.

In the first proposal, we will put together the knowledge obtained from the various firewall implementations presented in this study, in addition to other practices found elsewhere to design a general purpose firewall architecture that meets a wide range of security and functional requirements. In the second proposal, we discuss a special implementation of the general architecture to achieve multi firewall protection and authentication by using layered security levels.

6.1 General Purpose Firewall Architecture

Although, this proposed architecture focuses more on academic and research oriented security environments, it should be applicable to other non academic environments, where only the desired components can be installed. However, it should be emphasized that these components are generic. Therefore, the organization's needs, environment, and the

implications and complications of the services provided by the firewall should be studied first.

The firewall consists of several components that break the system into four security zones: (1) the secured network, (2) the screened subnet, (3) the exposed subnet, and (4) the untrusted network (the Internet), as shown in Figure 6.1. Each security zone has a different functional characteristics and therefore security requirements. Communications among these security zones takes place through various interface components of the firewall to provide the necessary security. A security policy is defined for these interfaces with the objective of achieving an acceptable balance between the supported services and the desired level of security within each security zone.

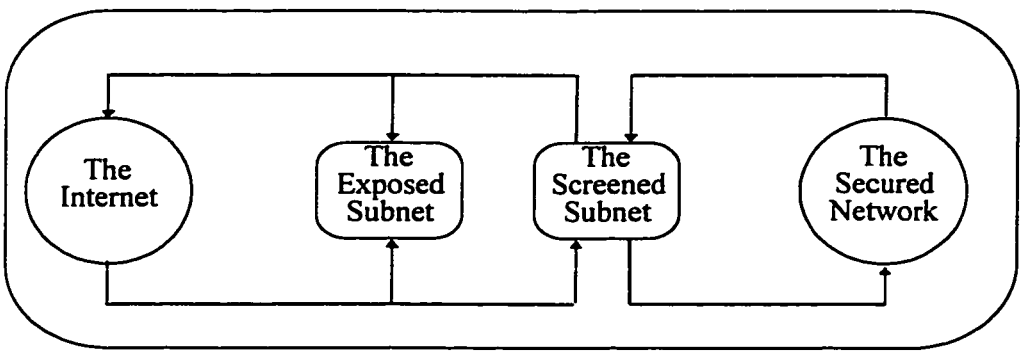


Figure 6.1. The four security zones.

The secured network contains the most vital information and resources. The proponent organization is normally interested in maintaining the integrity and confidentiality of such

information as well as in preventing the abuse of the available resources within this security zone.

The screened subnet is used to hide the secured network from other non secured networks. It is also used to provide the secured network with Internet services. Moreover, the screened subnet is to provide trusted external users with shared data and resources without having to get to the secured network.

The exposed subnet is a part of the private network that the proponent organization is interested in sharing with the rest of the world. It may contain services such as the FTP or WWW, that should be made available to the rest of the world.

The last security zone is the untrusted network (Internet). This is the world wide network that the organization is interacting with. Most of the security threats to the other security zones are attributed to the untrusted networks. Therefore, care and every effort should be made to establish a secure connection to it.

6.1.1 Network Users

Each of the four security zones houses a group of users that can be classified according to their influence on security. Before defining the firewall security policy, we should discuss the various types of network users in order to get a better understanding of the diverse needs of various security environments and the special operating requirements, such as special task

forces and working groups. It should be possible to control service availability to individual users as well as at the level of user groups. There are three major classes of network users.

The first class consists of the trusted network users. Trusted network users are defined as those authorized users to access and perform some operations on data and other resources in a given security zone. Trusted network users can be further divided into four subgroups as listed below:

1. Trusted users accessing the Internet from any host on the secured network.
2. Trusted users accessing the Internet from dedicated communication rooms on the secured network.
3. Trusted users accessing the secured network from a trusted host.
4. Trusted users accessing the secured network from other untrusted networks.

The second class is defined for the network guest users. The network guest users are those users who are authorized but with limited trust to access and perform some specified operations on data and other resources within a given security zone. This class consists of the following three subgroups:

1. Guest users accessing the Internet from inside the secured network.
2. Guest users accessing the secured network from a trusted host.

3. Guest users accessing the secured network from other untrusted networks.

The third class is made of untrusted network users. Untrusted network users are illegitimate and unauthorized crackers and intruders. This class of users poses the most security threats to information and other computing resources within the various security zones. This class consists of the following three subgroups:

1. Untrusted users trying to access the secured network from a trusted host.
2. Untrusted users trying to access the secured network from other untrusted networks.
3. Trusted users trying to perform some operations for which they are not trusted. This may include accessing the external network from the internal network.

Because, it is the user who makes things happen, the various classes and subgroups of network users have a great impact on the security policy. Therefore, the security policy should define what each type of users is allowed to do and where he can perform these actions.

6.1.2 Security Zones

As mentioned earlier, the four security zones: the secured network, the screened subnet, the exposed subnet, and the untrusted network (the Internet) have different functional and security requirements. In this section we will take a closer look at the most important security zones, namely, the secured network and the screened subnet.

6.1.2.1 The Secured Network

The driving force for the whole work found here and in similar implementations, is to maintain the integrity and confidentiality of the data and other information inside the secured network. Another reason that justifies the encountered cost and effort is the protection of internal computing facilities against abusive and unauthorized uses that may cause damage to these resources or denial of service.

The major security concern is related to outside users trying to break into the private network with good or bad intentions. Therefore, every effort should be made to keep unauthorized outsiders from getting to the protected resources. But unfortunately, there are sophisticated tools and determined crackers that can overcome most of the protection techniques. This results in limiting the provided services to a level that gives confidence in keeping intruders out of the secured zone. In some cases this may mean restricting legitimate users access to the protected zone from the outside.

The secured network should be completely isolated from other security zones, except through a single linking point that allows connections to be established from the inside to the outside. The reverse connection establishment, when allowed, should be carefully controlled and monitored. This can be achieved through the request-response technique, where TCP packets are not allowed to go into the secured network unless they are responses to requests that have originated from the inside. But for better security, reverse connection establishment may be completely disallowed.

For additional security and further controls over individuals and working groups, internal proxies are used to control who should access the Internet and how as will as what they should be allowed to do once they are connected to the Internet. Connectivity to the Internet, based on the type of user may be allowed from anywhere within the secured network or restricted to dedicated machines located in selected locations.

6.1.2.2 The Screened Subnet

The main objective of the screened subnet security zone is to hide and protect the secured network security zone from other security zones. All hosts within this zone are protected by the external screening router which makes the first level of protection. This first level of protection allows sharing of information and other resources with externally authorized users without endangering the secured network.

When external access is allowed to the secured network, the screened subnet is used as intermediate channel for all communication establishment and data transfers. Therefore, all Internet services that are designated to handle external connections are located within this zone.

6.1.3 Firewall Security Policy

The security policy for the proposed firewall implements the security policy that deny any thing which is not explicitly permitted. It defines what services the firewall should allow

in each security zone and what type of users should be permitted to perform actions there. It also defines what information and what type of users are allowed to cross from one security zone to another as well as the direction of the connection through various components of the firewall. The specific security rules are classified below:

1. Security Zones:

   a. Secured network:

      i) Confidential information and important data to the operation of the organization shou be kept inside the secured network.

      ii) Shared information and data with trusted external users should be kept inside the secured network.

      iii) The secured network should be isolated from other security zones, except through one single controlled connection.

      iv) The secured network should also be hidden from external networks.

      v) All outbound connections must be controlled at the level of user groups as well as individual users.

      vi) Inbound connection establishment to the secured network should be prohibited as much as possible. But, if necessary, it should be allowed through strong authentication and firewall filtering mechanism.

b. Screened subnet:

i) No user whether trusted or not, whether connecting from the inside or outside, is allowed to have log-in accounts in this security zone.

ii) No information or data should be kept in this security zone.

iii) The screened subnet should be used to hide the secured network from other security zones.

iv) All communication between the secured and untrusted networks must be made through servers on the screened subnet.

v) Inbound connection establishment is allowed to authorized users only.

c. Exposed subnet:

i) Any user whether trusted or not, whether connecting from the inside or outside, is allowed in this security zone.

ii) Guest users may be given log-in accounts on some dedicated workstations in this security zone.

iii) A copy of any information or resources that the organization would like to share with the rest of the world, should be placed in this security zone.

iv) It should be assumed that any information and resources in this security zone, are subject to corruption by intruders and crackers.

v) It should be possible to restore the state of information in this zone whenever is needed

or on regular basis.

d. Untrusted network.

i) Inside users should be able to access any untrusted network in the world without endangering the security of the internal network.

ii) Controls should be possible over connection establishment.

iii) External trusted and authorized users should be able to get to protected information at the right security zone without threatening the security of the system.

iv) External untrusted users should not be able to get into the protected security zones.

2. Service Access Policy: service access should be controlled at the level of user-groups and individual users.

6.1.4 Firewall Design Objectives

The design objectives of the proposed firewall include the following:

1. **Improve security:** In general, using the firewall should improve security and handle all known security holes. In addition to this general objective, one of the main objectives of the proposed firewall is to provide different levels of security to the information and to the users.

2. **Preserve services:** Usually, there must be a trade-off between security and services.

However, one of the design goals of this proposed architecture is to preserve the services of the system as much as possible while improving its security.

3. **Flexible Cost:** The proposed firewall architecture is flexible. It includes optional components that can be used or removed to achieve a balance between cost constraints and the minimum acceptable security and functional requirements as set by an organization.

4. **In-house development:** to help organizations in building their own firewall with the available limited resources and expertise.

6.1.5 Firewall Architecture

The proposed firewall architecture combines two commonly used firewall architectural solutions: the screened subnet architecture and the dual-homed host architecture.

In general, a screened subnet architecture, shown in Figure 6.2, is the most common do-it-yourself firewall architecture. It provides good security at a reasonable cost.

On the other hand, a dual-homed host architecture, shown in Figure 6.3, provides a lower-security at a lower-cost compared to the screened subnet architecture. The dual-homed host architecture is, therefore, often used by very small sites that are facing significant cost constraints.

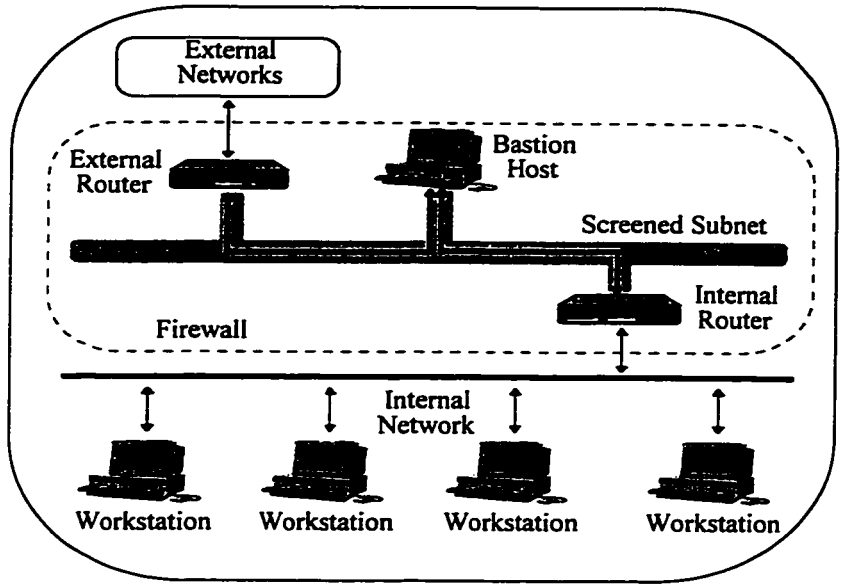


Figure 6.2. The Screened Subnet Architecture used with the proposed firewall architecture.

In our proposed solution as shown in Figure 6.4, we combine these two architectures to improve security while preserving services as much as possible at a moderate cost.

6.1.5.1 The Screened Subnet

The screened subnet can be used with single-router or two-router architectures, where, either a single three interface router or a pair of two-interface routers are used respectively. The single-router screened subnet architecture and the two-router screened subnet architecture work about the same. However, the single-router screened subnet architecture is cheaper, but the router must be able to handle both inbound and outbound packet filtering on each interface.

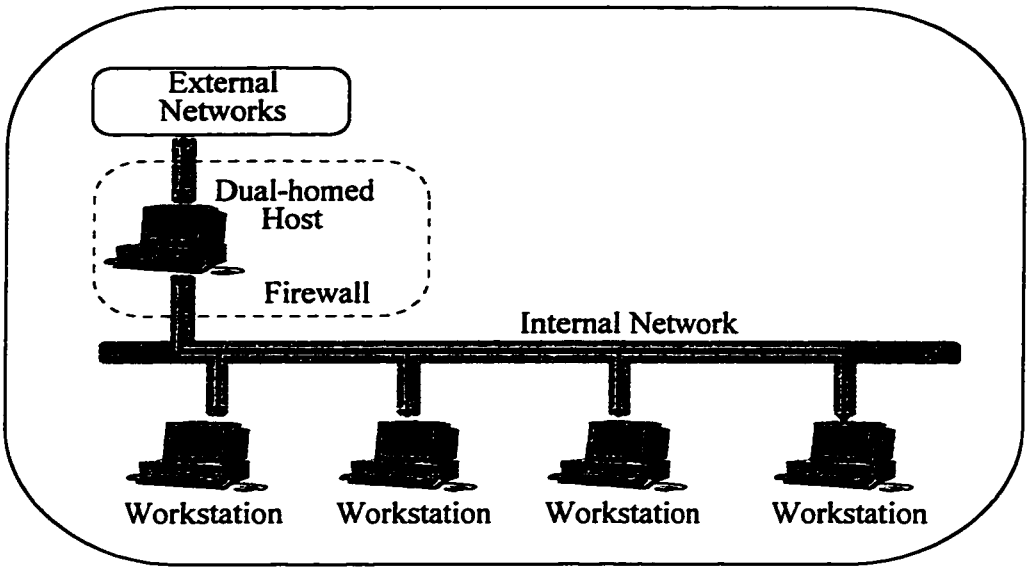


Figure 6.3. The Dual-homed Host Architecture used with the proposed firewall architecture.

6.1.5.2 The Dual-homed Host

The dual homed host can provide Internet services in one of two ways:

1. Having users login to the dual-homed host directly. If dual-homed host is the only protection for the network, user accounts present significant security problems. Because, users may unexpectedly enable services that are considered insecure. This threat can be reduced or eliminated by using other layers of security.

2. Providing services by proxying them. Proxying is less problematic, but may not be available for all services that are of interest to the site.

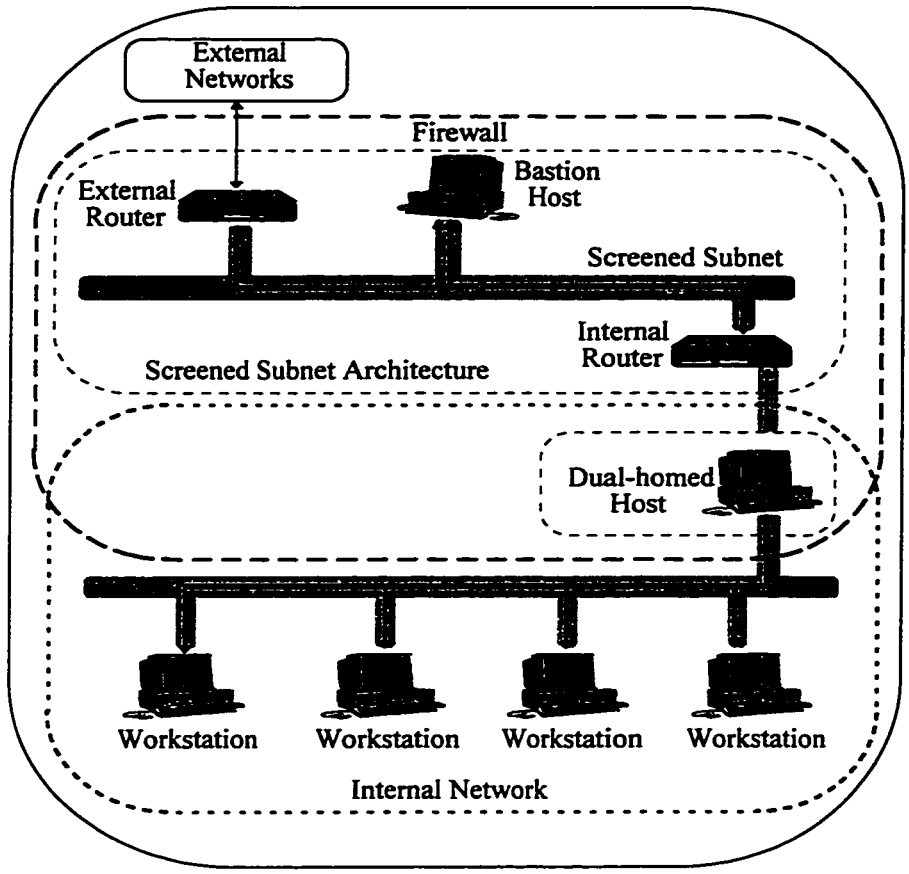


Figure 6.4. The Proposed Firewall Screening Architecture.

6.1.5.3 Assumptions

For the purposes of this proposal we will assume the following:

1. Internal users are not trusted and may actively try to circumvent the firewall, therefore, it is needed to monitor or log their Internet activities.

2. The use of IP addresses assigned to the site, and properly routed and advertised to the rest of the Internet by the service provider. Otherwise, proxies must be used, because

packets with unassigned IP addresses should not be allowed to go out onto the Internet. If they are allowed to do so, replies will have no way to come back to the site.

3. The use of separate network numbers, or at least subnet numbers, for the screened subnet and internal network. Using of separate network numbers makes it easier to detect forged packets [43] .

6.1.5.4 Functionality at the Component Level

Combining the screened subnet architecture with the dual-homed architecture results in considerable increase in securing the access to or from the Internet. This is due to the multi-layered protection.

In this architecture, the screened subnet provides: (1) First defense line, (2) Coarse filtering, (3) High level controls applicable to the whole organization.

The dual-homed hosts provide: (1) Second line of defense, (2) Fine filtering, (3) Segregation of users into groups of varying responsibilities and authorization.

Usually, there is a tradeoff between firewal cost and achieving better security while maintaining high levels of services. Therefore, two versions of the proposed firewall architecture we will presented.

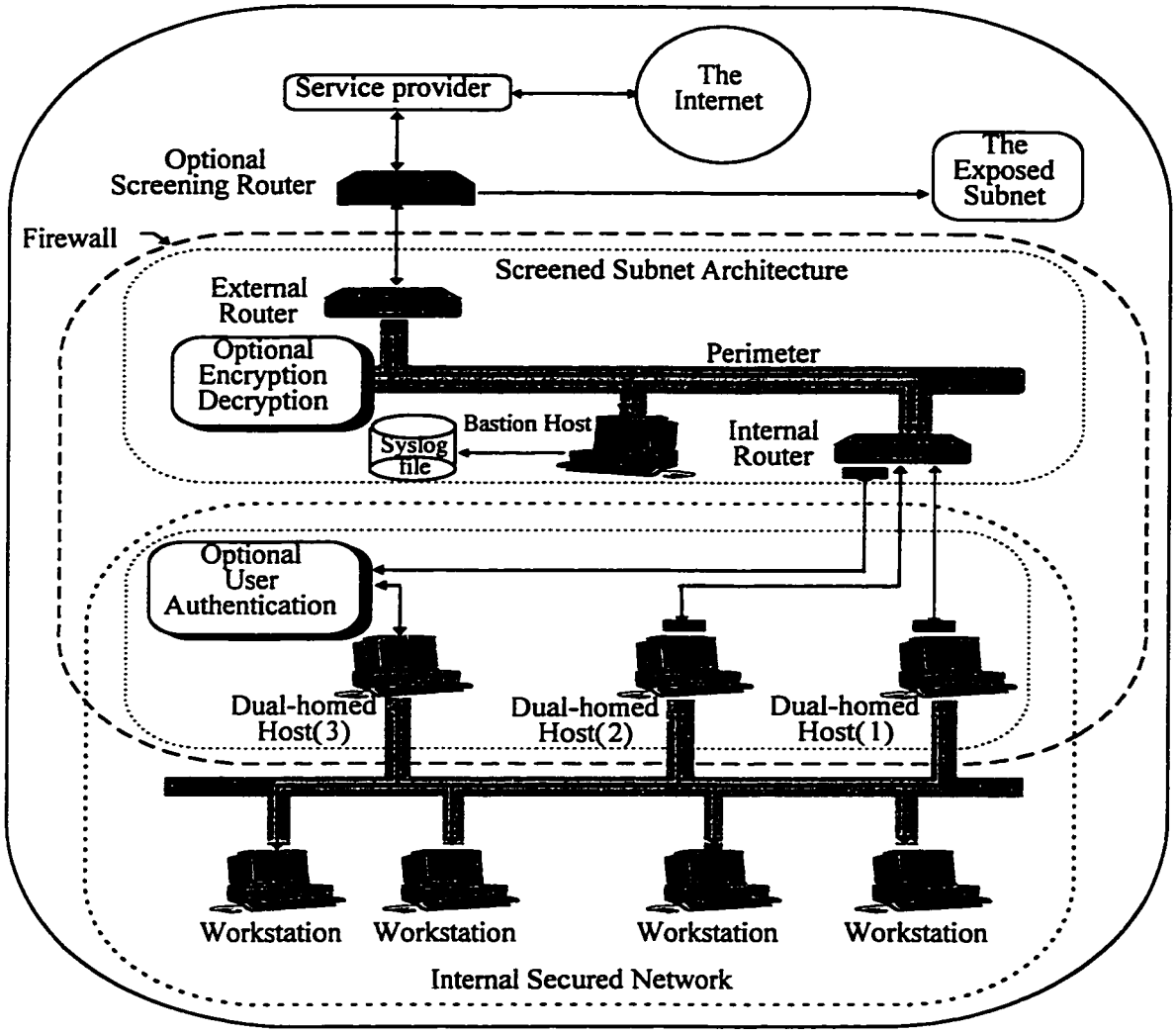


Figure 6.5. The Proposed Full Firewall Architecture.

The first version of the proposed firewall is a full architecture which is intended to achieve the best security controls while maintaining the highest possible level of services, as shown in Figure 6.5. In this architecture, the following components are used:

1. **The outermost router:** hides the internal network from the outside world and separates the traffic of the exposed subnet from that of the secured network. In this manner, the exposed network can not see any traffic addressed to the secured network. Therefore, reducing security threats to the protected network.

2. A two-router screened subnet is created between the secured network and other untrusted networks. The components of this type of screened subnet include the following:

    a. **Exterior router:** connects the site to the outside world and provides protection for the bastion host, interior router, and internal secured network.

    b. **Interior router**: Protects the internal secured network from the rest of the world. In addition, it isolates the site's own bastion host from the internal network, therefore, if a security breach is encountered on the bastion host of the screened subnet, the internal network will not be immediately affected.

    c. **Bastion host:** One or more bastion hosts can be used to serve as the site's main point of contact for incoming and outgoing connections from and to the outside world. The bastion host acts as proxy server for various services, this can be achieved in one of two ways:

i) By running specialized proxy server software for particular protocols, such as HTTP and FTP

ii) By running standard servers for self-proxying protocols, such as SMTP.

3. A set of dual-homed hosts on the internal secured network is used for internal and external users authentication as well as a client for various Internet services.

4. Optional security devices can be used to improve the security of the system when external trusted users are allowed to access the internal network from the Internet. Such security devices can be used to achieve the following two objectives:

   a. Devices, such as the SecurID can be used for a better user authentication as discussed in section 2.5.2.

   b. Devices, such as the Secure-Bridge can be used to improve the security when remote log-in is allowed. This goal is achieved by encrypting the information that is being exchanged between the site and other remote locations over the Internet.

5. In addition to the machines that make up the firewall itself, there should be one or more hosts on the internal network to handle internal services. Each of these internal services should be provided directly via packet filtering or indirectly via the proxy servers running on the bastion host of the screened subnet. The internal services include the following roles:

a. Mail server

b. Usenet news server

c. DNS server

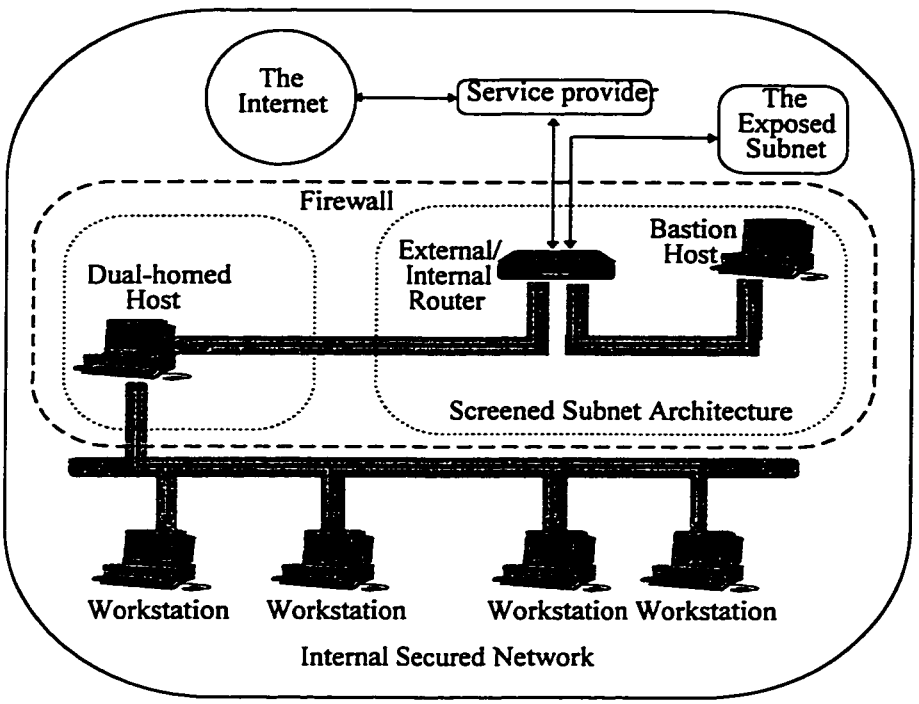


Figure 6.6. The Proposed Reduced Firewall Architecture.

The reduced version of the proposed firewall architecture decreases the cost of the proposed architecture by giving up some of its functionality while maintaining appropriate levels of provided services and security measures as much as possible. The reduced firewall architecture is shown in Figure 6.6. The following modifications can be considered when reducing the proposed firewall architecture:

1. The elimination of the outermost router. In this case, the traffic of the secured subnet will be visible to the exposed subnet.

2. The use of a screened subnet architecture with a single three interface router.

   The use of exterior routers that are owned and managed by the network service providers. But in this case, there will be a price to pay in terms of protecting the other components of the screened subnet against external hacking. This is due to the fact that the router will not be under the control of the site's own network administration.

3. The merge of the screened bastion host on the internal secured network and other internal services to act as a mail server, Usenet news server, and DNS server. This is in addition to its basic role of interacting with the screened subnet and authenticating internal and external users.

4. The elimination of the optional security devices and giving up some of the enhanced security measures. In this case, access from the Internet should not be allowed to the secured network at all.

6.1.6 Firewall Operation

The defined security policy is achieved via the implementation of the proposed firewall architecture. Where each component of the firewall architecture has a well defined security task as a part of the overall firewall security policy. In this section we will discuss the tasks

involved in building the firewall and how the various pieces fit together. Figure 6.7 gives an overview of the firewall, with the various security components that comprise the firewall and used to achieve the necessary security as defined by the security policy for each zone.

In the following discussion, we will explain how the basic Internet services are provided using the proposed architecture.

6.1.6.1 Outbound Traffic

Outbound services, from internal clients to servers on the Internet, are handled in either of the following two ways:

1. Allow internal clients to access external servers directly by setting up packet filtering on both the exterior and interior routers.
2. If proxy software is used on the firewall, allow internal clients to talk, by means of packet filtering, to the proxy servers running on the bastion hosts of the screened subnet and-vice versa. In this case, packet filtering should be set to prevent direct communication between internal clients and the outside world.

In both cases, the packet filtering allows the bastion host to connect to, and accept connections from, hosts on the Internet. The site's security policy dictates which hosts and what services are allowed through incoming and outgoing connections.

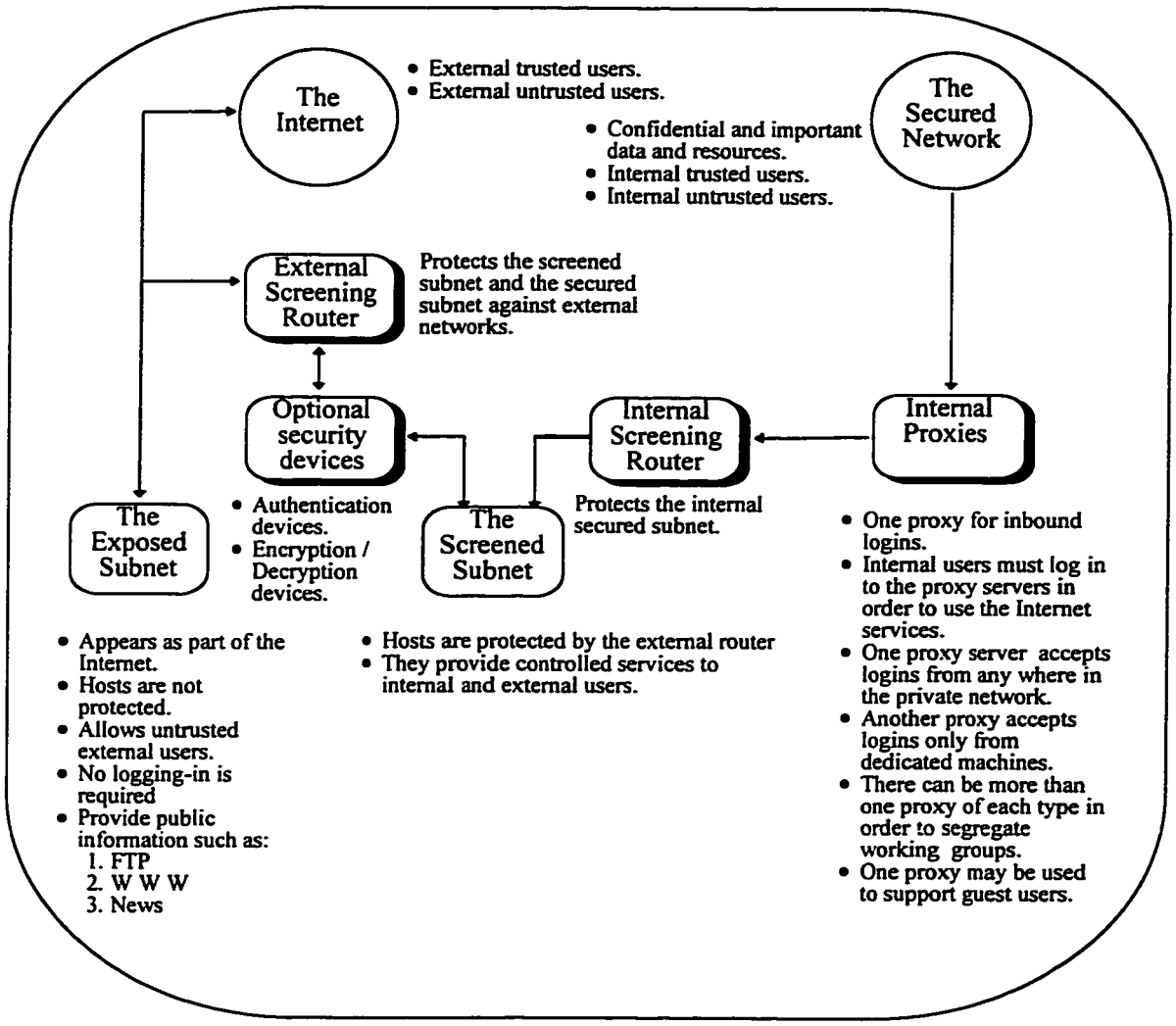


Figure 6.7. An Overview of the proposed Firewall Architecture

6.1.6.2 Inbound Traffic

Bastion hosts are the main point of contact for incoming connections from the outside world, such as: incoming e-mail (SMTP) sessions to deliver electronic mail to the site, incoming FTF connections to the site's anonymous FTP server, incoming domain name service (DNS) queries about the site, and so on.

On the bastion host of the screened subnet, a TCPWrapper daemon, which is hidden from normal users, can be used to start requested services. The TCPWrapper has the advantage of not requiring changes to existing software or configuration files. It is available by anonymous ftp from Edinhoven University of Technology. The TCPWrapper daemon perform its function by doing the following steps in order [48] :

1. Checks the client's address to see if it is as claimed. The daemon does this by checking the client's name against its trusted name-server.
2. If the above test is all right, the daemon checks the access control list and rejects the connection if the client is an unauthorized host.
3. In the system log file, through the system syslog facilities, it logs the accepted connections with a time stamp, the remote host address, and service requested.
4. Starts the real daemon.

6.1.6.3 Main Functions of Firewall Components

1. **Exterior router:**

   a. Directs all inbound traffic to the bastion host.

   b. Prevent unauthorized inbound services, such as Telnet and ping.

   c. Allows all outbound traffic. Optionally, it can be configured to block unauthorized outbound services in addition to the blocking done by the bastion host and/or the internal router.

2. **Bastion host:**

   a. Talks only to the Internal and external routers.

   b. For outbound traffic, it trusts a set of dedicate dual-homed hosts.

   c. Prevent unauthorized inbound and outbound services.

   d. For inbound connections, it checks user authorization and allow authorized users only. To achieve this task, a TCPWrapper daemon can be used to start other services when requested.

   e. For outbound connections, all users are considered trusted and allowed. The authorization for internal users is left to the dual-homed host on the secured subnet.

3. **Interior router**:

a. Accepts all outbound traffic from dedicated set of dual-homed hosts. Optionally, it can be configured to block unauthorized outbound services in addition to the blocking done by the bastion host and/or the external router.

b. Directs most of the outbound traffic to the bastion host.

c. Directs some of the outbound traffic directly to the external router, such a Telnet.

d. Prevent unauthorized inbound services, such as Telnet and ping.

e. Directs all inbound traffic to the dual-homed host that is dedicated to the authentication of external users and control of the allowable services to them.

4. **The set of dual-homed hosts:**

a. One separate dual-homed host is used for each group of internal users for the purposes of authentication and control of allowable services. Authorized internal users will have to logon to the designated dual-homed host before they can actually perform any communication with the outside world.

b. One dual-homed host is dedicated to authenticate external users and control allowable services to them. Authorized external users will have to logon twice, first to the designated dual-homed host, and secondly to the desired host on the secured subnet before they can actually perform any useful communication with the secured subnet. The purpose of the two stage logging is to hide the internally used logon IDs and pass-

words from the outside world.

6.1.7 Implementation

Due to the high cost in terms of hardware of implementing the fully proposed architecture (Figure 6.5), we have used the lab implementation of the test-bed to implement a slightly modified version of the reduced architecture (Figure 6.6). The only difference is the existence of an additional local area network behind the bastion host. This additional local area network fits the definition of the secured network zone. Therefore, resulting into two secured network zones.

The resulting setup is shown in Figure 6.8. While preserving all services and the principal idea, this architecture combined both internal and external routers and did not include any host on the exposed network.

By installing FWTK on the bastion host and SOCKS on the dual-homed host, we managed to have an operating firewall mechanism identical to that of the reduced version of the proposed architecture.

6.1.8 Possible application of the Proposed Architecture

Using the architecture of Figure 6.8, a double firewall protection can be achieved by (1) configuring SOCKS to allow direct outbound connections, while, not accepting any inbound connections except from FWTK bastion host. (2) configuring FWTK to allow users to login

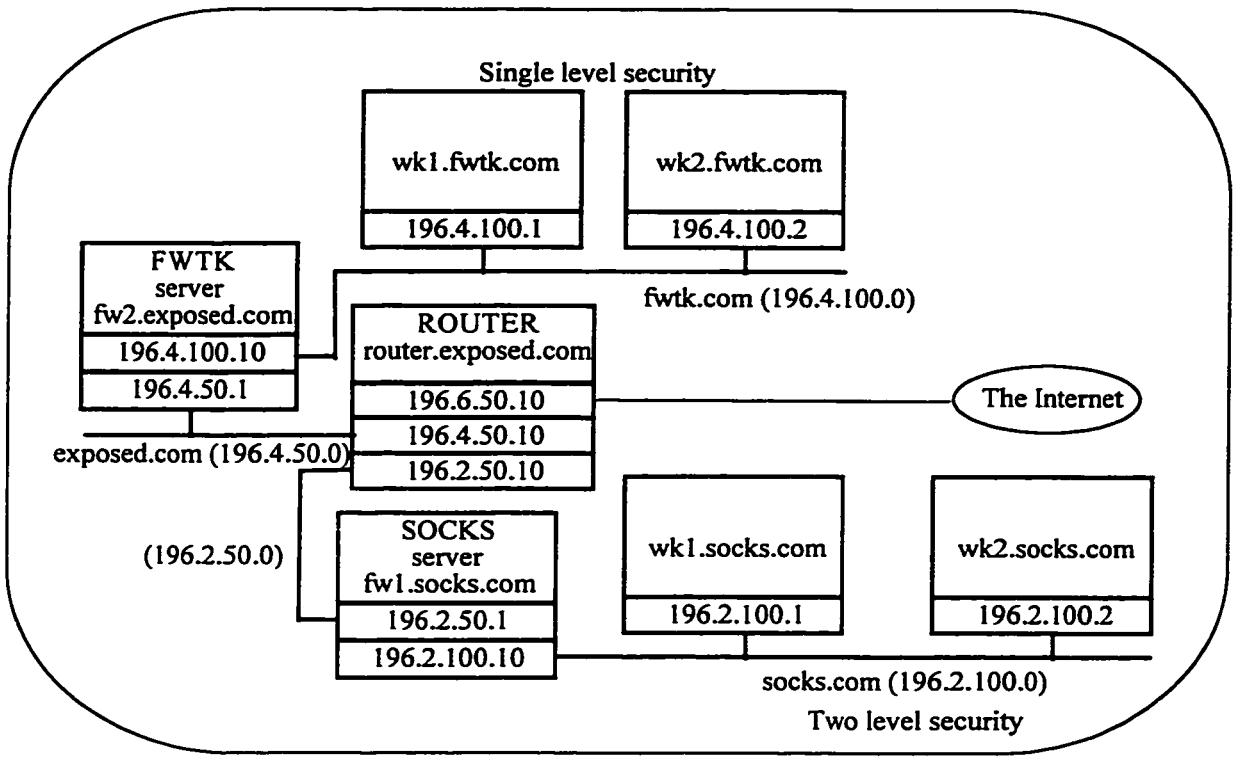


Figure 6.8. Lab Implementation of the reduced version of the proposed architecture on the selected Test-Bed.

to the bastion local host and using the Unix command "chroot" to lock the user into a locked-room directory. (3) providing users with socksified services in the locked room directory. In this manner, while internal users can directly connect to the Internet upon successful authentication to SOCKS, external users will have to follow the procedure outlined below before they can successfully connect to an internal host:

1. Authenticate them selves to the FWTK.

2. Connect to the FWTK bastion host as local users, where, they will be automatically

locked into a locked room directory.

3. Using the allowed socksified services in the locked room directory, users can then authenticate themselves to SOCKS to connect to the desired host behind it.

6.1.8.1 Problems and Assumptions

The TIS FWTK is designed so that it would be possible to use the FWTK "netacl" to make inbound connections to the FWTK firewall server using the tn-gw proxy which is included with the FWTK. Where, connections to the FWTK "localhost" using the standard telnet service should be possible afterward. We have been able to use the tn-gw by it self to connect to the FWTK server, but when used in combination with the FWTK "netacl", the connection to the FWTK firewall is closed, therefore, failing to complete the first stage of the two level connection procedure.

In this experiment, it is assumed that the external router can filter based on source, destination, service port, and inbound versus outbound connections. In addition, we assumed that, it is possible to use the FWTK "netacl to make inbound connections to the FWTK firewall server using the tn-gw proxy which is included with the FWTK. After that, it should allow connection to the FWTK "localhost" using the standard telnet service.

The disadvantage of this approach is the need to login to the FWTK firewall "localhost" which increases the risk of compromising the FWTK firewall. Normally, this is considered to be dangerous, but the risk can be greatly minimized through the use of strong authentication.

In addition, through the use of the "locked-room" directory concept, the damage can be minimized in case of compromising the FWTK firewall. Therefore, there is a need to design a better architecture that will provide multi level protection without reducing the security of the bastion host. In the following section we will discuss a new proposal that can achieve this goal.

6.2 Multi Level Firewall Protection

Multi firewall protection which includes multi firewall authentication is a special implementation of the proposed general firewall architecture using layered security levels. In this approach one or more firewall packages are used at each security level to satisfy different requirements and to provide different services. Firewalls used on the same or different security layers can be of the same or different types depending on security and functionality requirements.

As shown in Figure 6.9, the reduced proposed firewall architecture is modified to provide the basic firewall architecture for implementing the proposed multi level firewall protection technique. The modified architecture allows one more additional layer of security which fits the definition of the secure internal security zone. It also allows safe connection through the internal firewall after a successful authentication by the external firewall. The resulting security layers are:

1. **Exposed security zone**: The exposed security zone is connected to the external router.

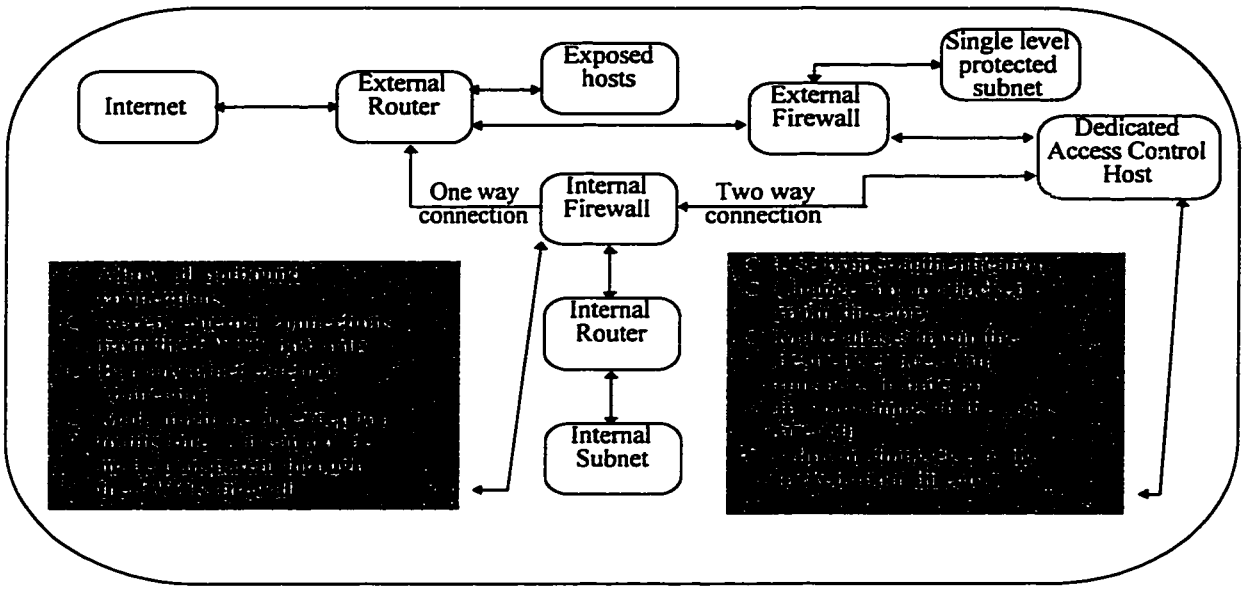


Figure 6.9. Double Firewall Architecture

Authentication is not required to access information on this subnet. The only protection that is available for the exposed security zone subnet against external attacks is provided via the external router.

2. **One level security zone**: This is a single level protected network. It is protected by the external firewall behind the external router. Authentication by the external firewall is required to connect and access information in this zone.

3. **Two level security zone**: The two level security zone is in fact the Internal subnet. Inbound connections to the internal subnet are checked by the external and internal routers, the external and internal firewalls respectively. While, the outbound connections are authenticated only by the internal firewall and sent directly to the

external router.

In this setup the external router will drop any inbound connection that is not addressed to an exposed host or to the external firewall bastion host. While it will accept all outbound connections originating from the one or the two level security zones.

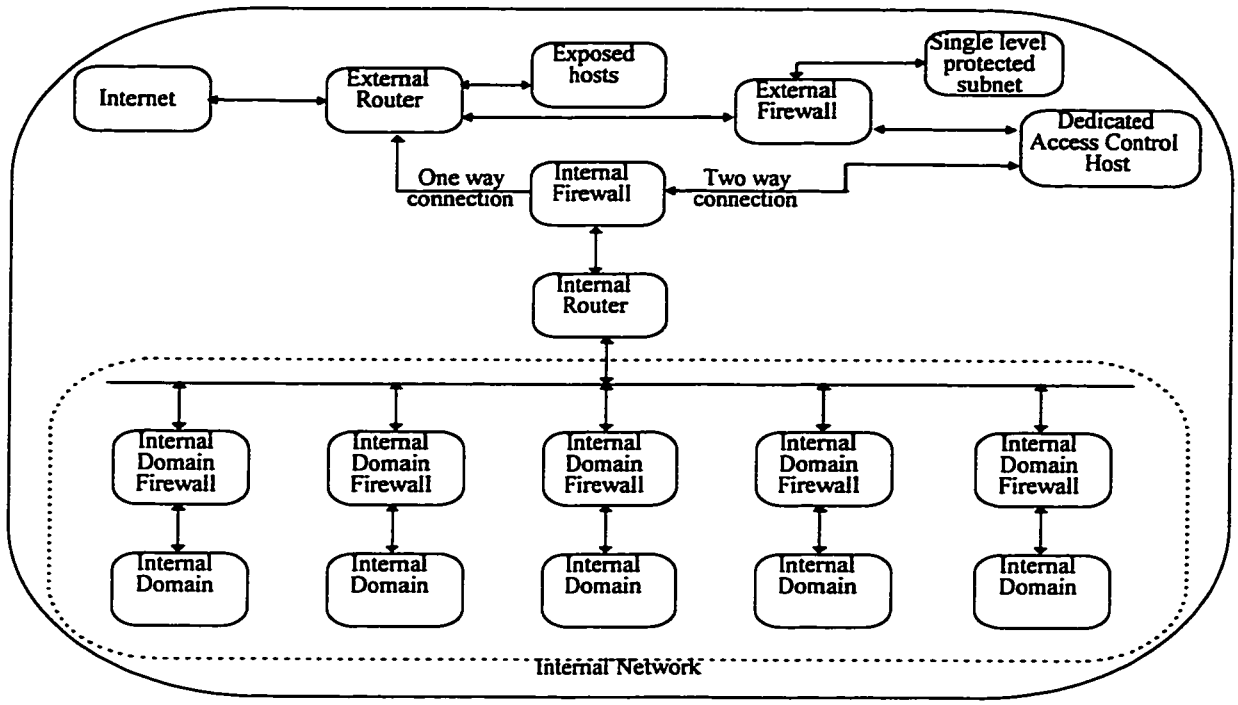


Figure 6.10. Three Level Firewall Architecture

There are situations where internal administrative domains need to install internal firewalls to restrict access from other internal administrative domains. This objective can easily be achieved as shown in Figure 6.10 by expanding the double firewall architecture into a three level firewall architecture. In this manner, the main internal firewall at the second level is used as the main entry point to the internal network. It transparently connects external

users as well as users of other domains to the desired internal domain firewall. While, the main internal firewall can be used for an overall authentication and security controls, each internal domain firewall can be used to impose specific security controls.

6.2.1 Principal Idea

In general, the principal idea of multi level firewall protection is that in case of compromising one level of protection, the next level of protection will continue to protect the secure network. Also, the use of hardware routers compared to software routers is considered to be more secure because it is more difficult if not impossible for a hacker to manipulate the routers routing tables. However, in this architecture we have the following additional features:

1. When the external protection is compromised, internal protection mechanism will not be directly exposed to external attacks.
2. In case of external firewall failure, internal users will not loose the ability to connect to the Internet.
3. Normally, outbound connections are much more than inbound connections. Therefore, because less check points are used with outbound connections compared to inbound connections, the effect of multi firewall protection on communication speed is reduced.

In this section we explain this technique by means of an example using the two firewalls that we examined in chapter 5, namely SOCKS v5 and TIS FWTK 2.0. In practice[15], both TIS FWTK and SOCKS have been used simultaneously to provide a secure Internet connectivity. TIS FWTK is used for inbound traffic requiring strong authentication on telnet and ftp, where people are forced to do some 'extra work' to enter the protected network from the Internet. On the other hand, SOCKS is used for outbound traffic, to keep the use of telnet, ftp, and other services as transparent to the internal user community as possible without requiring any additional typing.

6.2.2 Implementation of the Double Firewall Protection

In this example we will combine the two firewalls to create a two level defense perimeter. Combining the two firewalls will improve Internet connection security, functionality, and convenience of internal users. This is made possible through the cooperation of both firewalls to share their pros and compensate for each other cons. As shown in Figure 6.11, two routers, external and internal, are used, where:

1. The external router sends all inbound connection requests to the FWTK firewall server.
2. The internal router:
    a. Sends all outbound requests to the external router

[15] According to some questions and answers found over the Internet. These Internet messages can be searched for under various topics that compare SOCKS and TIS FWTK such as "SOCKS vs. TIS" and "SOCKs vs FWTK".

b. Accepts inbound requests only from the FWTK firewall server.

3. The FWTK firewall server:

   a. Authenticates external users before they can access the internal network.

   b. Is an application level firewall type, therefore, it provides higher level filtering.

   c. Allows, only, services with installed proxies.

4. The SOCKS firewall server:

   a. Is a circuit level firewall type which provides transparent check point.

   b. It allows any allowed socksified services to go through.

   c. Controls internal users' access to the Internet.

   d. Accepts external access only from the dedicated access control host behind the FWTK firewall server.

In this experiment, all outbound connections will pass through the SOCKS firewall transparent to internal users, while, the SOCKS firewall will not accept any external inbound connections not originating from the dedicated access control host behind the FWTK firewall server.

All inbound connections to any host located in the two level security zone will be done in two stages depending. In the first stage, external users are required to connect to the

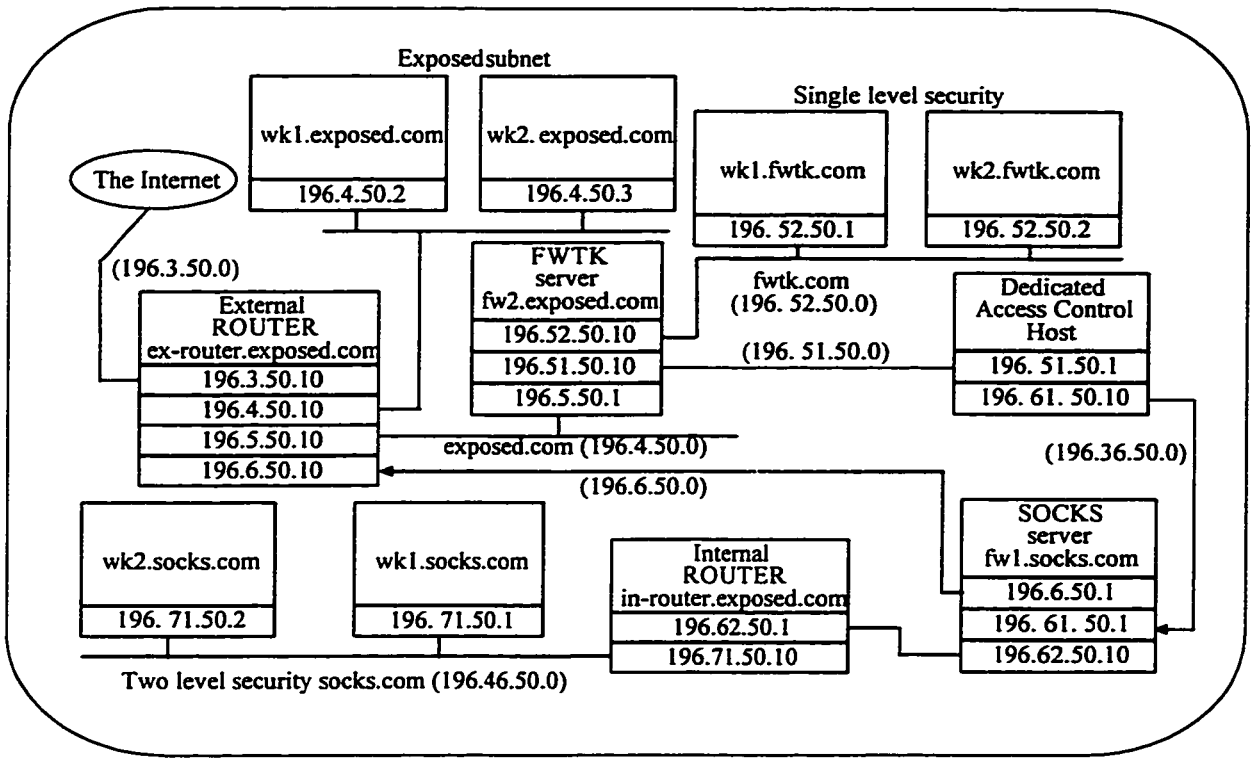


Figure 6.11. Example Implementation of the Combined SOCKS and FWTK firewalls

dedicated access control host behind the FWTK, where simple or strong authentication can be used. At this stage, successfully logged on external users, depending on the configuration, may be locked into a "locked-room" directory via a "chroot" command. The second stage can be executed only by authorized external users after succeeding in connecting to the dedicated access control host behind the FWTK firewall in the first stage. An authorized user, in the "locked-room" directory, can use the available socksified services to connect through the SOCKS firewall server into the two level security zone.

This implementation of the proposed double firewall architecture using SOCKS and FWTK provides the following benefits:

1. Since SOCKS firewall is of circuit-level firewall type running at the transport layer and since most connections are outbound connections, this implementation will have a minimum effect on communication speed.
2. Transparent use for internal users
3. External users think there is only one firewall (FWTK).
4. Two levels of security with different security controls.
5. While the FWTK firewall is directly exposed to external users, the SOCKS firewall is hidden and protected by the external firewall (FWTK).
6. External users must authenticate themselves to the external firewall before they can get through the internal firewall.
7. Supports the proposed security zones as explained in section 6.1.2.

6.2.3 Comparative Analysis of Strengths and Weaknesses

It is a well known fact that there is no completely secure lock. But, using locks makes it more difficult to thieves to break into protected properties. Similarly, security tools in computing environments do not provide unconditionally secure protection. Therefore, any

security measures are expected to have some advantages as well as some disadvantages and limitations of what it can, and can not do. The proposed double firewall architecture is no exception of this fact. In this section we will analyze its strengths and weaknesses.

SOCKS is a circuit level firewall which is faster than application level firewalls such as the TIS FWTK. Therefore, outbound traffic should not have much of a bottleneck problem. On the other hand, inbound traffic will have to go through TIS FWTK firewall in addition to the SOCKS firewall. This is expected to slow the inbound traffic down. However, this should not be a major problem since external connections to the internal network are expected to be at minimum to reduce security risks. Naturally, it is far more important to ensure security at the account of some user convenience.

Comparing the implementation of our proposed firewall architecture, using SOCKS and FWTK firewall packages, to the SURF firewall (see section 4.3), the SURF firewall has some common functional requirements with our proposed firewall. In the design of the SURF firewall, there are three potential sources of vulnerability. These are (1) the open environment behind the firewall and absence of outgoing packet filtering, (2) the coarseness of the incoming packet filter, and (3) the potential for hijacking acceptable connections. We will discuss below how these vulnerabilities are addressed using our proposed firewall.

6.2.3.1 Open Research Environment

In the SURF firewall, the first risk arises because the traffic between machines behind the firewall is not restricted and because no outbound packets are filtered. As a result, if an intruder penetrates the security perimeter, he will have unrestricted access to all internal hosts. Once an intruder gains access to an internal host, the firewall no longer offers any protection, since SURF does not prevent any outbound operation, including data transfer.

In our proposed firewall architecture, we believe that we have succeeded in limiting these risks without significantly affecting the openness of the network environment. The reasons behind this are:

1. The internal secured network is protected by more than one layer of the proposed firewall architecture.
2. It is harder for an outsider to break into the internal secured network, because he will have to go through multiple stages of authentications. An intruder has to compromise all protection levels, otherwise he will not gain any thing.
3. Even if an intruder manages to gain external access to the system, he will not be automatically granted internal user privileges, since, the outbound traffic is filtered transparently to the user. Moreover, internal users may be required to authenticate themselves for the type of outgoing services.

6.2.3.2 Coarse-Grain Packet Filter

In the SURF firewall, the second risk comes from the use of a coarse-grain packet filtering while relying on internal hosts to perform additional filtering. As a result, the packet filter admits packets that may not be responses to outstanding requests. This may expose the internal network to a denial-of-service attack flooding the network with extraneous packets that internal hosts must process and drop.

In our proposed firewall architecture, this problem will be of less significance since only valid packets are admitted.

6.2.3.3 Connection Attacks

The last potential risk arises from attacks to authorized connections through the firewall, such as TCP sessions to external hosts. The SURF firewall is vulnerable to this kind of attack because it supports authenticated access through potentially insecure hosts and networks outside the firewall. For an example, NFS requests to a compromised external file server may cause the firewall to read or execute intruder-supplied data.

To address these dangers in our proposed firewall architecture, care should be taken in selecting applications to be made available to users. For example, applications with response packets that could obtain control of a shell running inside the firewall, should not be permitted through the firewall.

6.2.4 Possible Improvements

The proposed firewall architecture does not work by itself, rather, some firewall software package or packages need to be installed and used with it. In our implementation of the proposed firewall architecture, we have used SOCKS and FWTK as the two firewall software packages to have a working firewall system.

Therefore, one possible improvement to this proposed firewall system can be achieved by modifying the FWTK components so that they can talk to the SOCKS firewall on one side of the FWTK, while talking to the rest of the world on the other side. The advantage of such modification is that there will be no need to use a dedicated access control host, therefore, reducing the cost of hardware. Also, with such modification, it should be possible to keep the same functionality with only one stage logging.

Another possible improvement involves customizing this proposed architecture to extend offered firewall protection to other sites which use wireless communication.

6.3 Comparison With Other Firewall Architectures

In section 3.3 we discussed the three main firewall architectures. Using each of these architectures separately provides single level protection, where, the same route is shared by the firewall for both inbound and outbound connections.

In addition we have explained how variations of these architectures can be utilized to generate other acceptable firewall architectures. We have used one of the acceptable variations, "combining screened subnet with dual-homed architectures", to design the proposed firewall architecture. We have also used another acceptable variation, "combining external internal routers", to reduce the cost of the proposed architecture.

Comparing the proposed architecture to other firewall architectures as discussed in section 3.3, we can see that it provides additional features and advantages which include:

1. Generality to meet wide range of functional and security requirements.
2. Expendability to accommodate multi-level firewall protection.
3. Flexibility to accommodate variable cost implementations.
4. Possibility to provide separate filtering routes for inbound and outbound connections. This provides two advantages:
    a. Additional special purpose hardware/software such as encryption /decryption or authentication can be placed on one route but not the other.
    b. Minimal effect on performance can be achieved by reducing the number of chick points on the heavily used route which is normally the outbound route.
5. Distribution and classification of security controls:

a. Global controls and coarse grain filtering are provided at the highest level by the screened subnet.

b. Customized controls and finer filtering are provided at lower levels by individual dual-homed hosts.

c. Segregation of users into groups off similar authentication and common service requirements.

6.4 Summary

In this chapter, we have proposed a general firewall architecture that meets a wide range of security and functionality levels, as well as, varying cost requirements. Then we have shown how it can be used to achieve multi level firewall protection. This approach was illustrated using SOCKS and TIS FWTK firewalls. Finally we compared the proposed firewall architecture to other firewall architectures.

CHAPTER 7

CONCLUSION

Different methods have been in practice to achieve network security. However, with the rapid expansion of the world's largest computer network (the Internet), by connecting more and more computer networks throughout the world, computer security became an important issue. Especially, against unauthorized access which facilitates resource abuse and threatens data secrecy and integrity. Therefore, new security techniques should be used to protect local networks against intrusion from the Internet. Basically, we need to prevent destruction of data by intruders, maintain the privacy of local information, and prevent unauthorized use of computing resources.

To improve network security, network connections to the Internet, in general, do not take place transparently. Instead, firewall servers are used to protect the systems connected to the local network against assaults from the Internet. Firewalls are considered to be one of the best and most reliable means of network protection against intruders. But, there is a price to pay, usually, because the firewall server results in a bottleneck for assaults from the Internet into the LAN as well as for allowed communication between the LAN and the Internet.

Firewalls can be valuable resources when implemented in a security context for host computers, LANs, and Internet connectivity. Firewalls are specifically designed to protect Internet-linked networks against unauthorized access. They work by filtering packets and handle a variety of protocols. Many firewalls provide features such as network address translation, authentication and virtual private networks. Most of these firewalls fall into three classes or firewall types. These types are router-based, circuit-level gateways, and application level gateways. In this work we have examined two of the most popular firewall products (SOCKS and TIS FWTK) using some security evaluation tools (SATAN). We have designed a layout for testing methodology and investigated the possibility of customizing the tested firewall products.

We expect the demand for firewalls to be increased because of the ongoing threat of intruders, rapid adoption of Internet strategies, and increased need for internal security.

7.1 Thesis Summary

In this thesis, intensive literature review in the area was done. Firewall background information was studied which included the study of Internet related security concerns, the TCP/IP protocol used in Internet communication, security analysis tools with SATAN as an example, different protection methods, and various security policies. Firewall overview and relation to the overall security policy was presented. In addition, firewall types and

architectures were studied in more details. For illustration, some firewall examples were selected to give an insight to the wide range of firewall design considerations and principle.

We have, also, designed varying cost test-beds to be used in firewall testing and evaluation. A testing methodology was defined to be used along with the selected test-bed. The defined methodology was applied to test and evaluate SOCKS and TIS FWTK firewalls using SATAN. Test results were reported in two classes, automatically generated and manually observed. Finally, new proposed firewall solutions were presented and discussed in detail.

The result of evaluating the two firewalls does not favor one over the other. However, it indicates that each firewall suits different environment requirements. For instance, the FWTK firewall, being of the application level type, provides more security and less flexibility which suits corporate more than academic security environments. On the other hand, the SOCKS firewall, being of the circuit level type, provides more flexibility and less security which suits academic more than corporate security environments.

Finally, we have proposed a general firewall architecture that meets a wide range of functionality and security requirements. Some of its discussed implementations illustrated how the proposed firewall architecture can be used with SOCKS and FWTK to achieve multi firewall protection.

7.2 Possible Applications

The work done in this thesis consists of two parts: 1) Testing and Evaluation of firewalls and 2) Proposal of new firewall solutions.

The first part has its application in organizations planning to connect to the Internet or use firewalls of some kind to setup an Intranet or an Inter-Company network. Where, it becomes important to select a firewall that meets the organization's security and functional requirements. Before, deciding which firewall to use, several firewalls may be tested and evaluated using one of the test-beds and the testing methodology as presented in this theses.

The second part can be used to design a varying cost firewall architecture that meets wide range of functional and security requirements.

7.3 Future Research

In this thesis we have used SATAN security analysis tool to perform the automated evaluation of the two firewalls. SATAN, at the time when we started this project, was the most current and widely publicized. But, as of now, most of the security holes that it checks have been fixed in recent software. New vulnerabilities may have been introduced with new software, therefore, in similar future work, new and more powerful tools need to be investigated. Such tools should be capable of exploiting vulnerabilities on the firewall to reach protected machines for more security analysis.

When using the proposed multi firewall architecture with SOCKS and TIS FWTK, a possible improvement can be achieved by modifying FWTK proxies and/or the SOCKS firewall to facilitate connections through their combined firewall. By doing so:

1. There will be no need to dedicate one workstation for log-in to the SOCKS firewall, therefore, reducing the hardware cost.
2. The same functionality can be achieved with only one stage logging.

Another possibility that is worth investigating is using the FWTK in place of SOCKS to provide tunneling that may provide functionality similar to the transparent circuit level service of the SOCKS firewall.

Also, sharing one bastion host to run both firewalls will facilitate using socksified clients wherever possible. While, hosts that do not have such clients still can use the non-transparent proxies of the FWTK.

A better improvement to the proposed multi level firewall architecture can be achieved by developing an integrated two-part software package. The first part of the firewall replaces the FWTK firewall and supports application level firewall type. The second part replaces the SOCKS firewall and works transparently as a circuit level firewall type. The two parts of the firewall should be able to communicate with each other as well as external SOCKS and any standard services firewall.

APPENDIX A
Related Definitions

Administrative domain: A group of computers that belong together for administrative purposes. **Domain Name Server (DNS):** A computer server that is designated to translate the meaningful names into numerical Internet addresses.

IP Datagram: The basic unit of information passed across the Internet. It contains source and destination addresses together with data and a number of fields that define other information such as the length of the datagram, the header checksum, and flags to say whether the datagram can be or has been fragmented.

Packet: Packets are the fundamental units of communication on the Internet. A packet is a defined format of data which is sent over the Internet, where each packet is constructed of a header, a source and destination addresses, and data fields.

Port: The concept used by Internet transport protocols to distinguish among multiple simultaneous connections to a single destination host.

TCP: Transmission Control Protocol. It is the major transport protocol in the Internet suite of protocols that provides reliable, connection-oriented, and full-duplex streams using IP for delivery. The TCP is a connection-oriented protocol, unlike the connectionless IP, it provides the means to associate specific communicating end nodes [15] .

Telnet: Telnet is a popular application protocol used for remote-system configuration and management. It is also a favorite tool for hackers.

UDP: User Data Protocol is one of the Internet transport protocols. UDP, like TCP, uses IP for delivery, but, unlike TCP, UDP is connection-less protocol and thus allows exchange of datagrams without acknowledgments or guaranteed delivery.

UNIX: A popular computer operating system which is condidered to be the primary operating system on the Internet that provides multiple clients with simultaneous access from network locations.

Virtual Private Networks: Multiple sites on the Internet having private communications.

APPENDIX B
Useful Scripts and Aliases

To facilitate switching between the different test conditions, the following scripts are used to automate the switching process. These scripts require saving two copies of LINUX kernels (one with enabled IP forwarding and one with disabled IP forwarding). For easiness and more transparency some aliases are defined as shown below:

- Enable-ip-forwarding:

```
#!/bin/sh
# This script installs a kernel with IP forwarding enabled
cp /vmlinuz.with.ip.forwarding /vmlinuz
lilo
echo 'forwarding=enabled' > /firewalls/scripts/forwarding.stat
shutdown -r now
```

- This script enables IP forwarding as follows:

  * The script will copy the **/vmlinuz** kernel from **/vmlinuz.with.ip.forwarding** which has been previously configured to enable IP forwarding,
  * Executes the linux loader "lilo",
  * Stores the state "enabled" of the IP forwarding in "/firewalls /scripts /forwarding.stat", to help the administrator in knowing the status of IP forwarding at any time,
  * Finally, the script will restart the system, in order to activate new kernel.

- Disable-ip-forwarding

```
#!/bin/sh
# This script installs a kernel with IP forwarding disabled
cp /vmlinuz.no.ip.forwarding /vmlinuz
lilo
echo 'forwarding=disabled' > /firewalls/scripts/forwarding.stat
shutdown -r now
```

- This script disables IP forwarding as follows:

  * The script will copy the **/vmlinuz** kernel from **/vmlinuz.no.ip.forwarding** which has been previously configured to disable IP forwarding,

  * Executes the linux loader "lilo",

  * Stores the state "disabled" of the IP forwarding in "/firewalls /scripts /forwarding.stat", to help the administrator in knowing the status of IP forwarding at any time,

  * Finally, the script will restart the system, in order to activate new kernel.

- Services-max

```
#!/bin/sh
# This script sets the full services
clear
cp /etc/inetd.conf.max /etc/inetd.conf
cp /etc/services.normal /etc/services
ps -x | grep inetd
echo 'services=Maximum' > /firewalls/scripts/services.stat
echo '*----------( Maximum Services )----------*'
echo 'Please type the following command:'
echo 'kill -HUP <==the proceess ID of the inetd as shown above'
echo '*----------*'
```

- This script sets the full services as follows:

* The script will copy the "/etc/ inetd.conf" from "/etc/ inetd.conf.max" which has been previously configured to enable all standard services,
* The command "cp /etc/services.normal /etc/services" shown here has no effect, because, as it is shown, it only copies the normal services file. However, it is needed in the case when new ports are defined to meet some TIS FWTK configuration requirements. In this case, there will be two services files one for the normal set and the other one for the TIS FWTK set,
* The command "ps -x | grep inetd" is used to get the process Id. of the **Inetd** daemon
* Stores the services state "Maximum" in "/firewalls/ scripts/ services.stat", to help the administrator in knowing the status of IP forwarding at any time,
* Finally, the script will instruct the administrator to execute the command "kill -HUP" using the process Id. of the **Inetd** daemon, to instruct **Inetd** to reread its configuration file without rebooting the system.

- Services-min

```
#!/bin/sh
# This script sets minimum services
clear
cp /etc/inetd.conf.min /etc/inetd.conf
cp /etc/services.normal /etc/services
ps -x | grep inetd
echo 'services=minimum' > /firewalls/scripts/services.stat
echo '*————————————( Minimum Services )————————————*'
echo 'Please type the following command:'
echo 'kill -HUP <==the proceess ID of the inetd as shown above'
echo '*————————————————————————————————————————————*'
```

- This script sets the minimum services as follows:

  * The script will copy the "/etc/ inetd.conf" from "/etc/ inetd.conf.min" which has been previously configured to only enable the minimum necessary standard services,

  * The command "cp /etc/services.normal /etc/services" shown here has no effect, because, as it is shown, it only copies the normal services file. However, it is needed in the case when new ports are defined to meet some TIS FWTK configuration requirements. In this case, there will be two services files one for the normal set and the other one for the TIS FWTK set,

  * The command "ps -x | grep inetd" is used to get the process Id. of the **Inetd** daemon

  * Stores the services state "minimum" in "/firewalls/ scripts/ services.stat", to help the administrator in knowing the status of IP forwarding at any time,

  * Finally, the script will instruct the administrator to execute the command "kill -HUP" using the process Id. of the **Inetd** daemon, to instruct **Inetd** to reread its configuration

file without rebooting the system.

- Aliases.txt

```
alias runsocks='/usr/local/socks/runsocks
alias ftp='runsocks ftp'
alias telnet='runsocks telnet'
alias rlogin='runsocks rlogin'
```

- This script sets the following aliases:

  * "runsocks" to conveniently starts the socks firewall daemon without having to be in the socks directory.

  * "ftp", "telnet", and "rlogin" to transparently socksify and use the corresponding standard service through the SOCKS firewall. This will enable users to use the socksified services as if they are the standard services.

- Unaliases.txt

```
unalias runsocks
unalias ftp
unalias telnet
unalias rlogin
```

- This script is used to cancel aliases that were set using the above script. This will enable users to use the normal standard services.

APPENDIX C
Recommendations

The following recommendations should help organization in achieving their network security using firewalls. These recommendations are aimed toward preparing the network environment for the use of firewall protection [3] :

1. **Standardization:** Operating system and other software should be standardized in order to make installations of new programs and other security related fixes more manageable.
2. **Procedures:** A procedure should be established to achieve efficient, site-wide installation of programs and new software.
3. **Proper tools:** The use of proper utility programs and services that will assist in achieving centralized system administration, should be considered if centralized system administration will result in a better security and administration.
4. **Preventive security checks:** Host systems should be periodically scanned and checked for common vulnerabilities and configuration errors.

APPENDIX D
Securing Services With Firewalls

One of the elements that binds the Internet together is a set of Unix-based protocols that let us access files and locate and use resources at remote sites. Some of these services are File Transfer Protocol (FTP), Domain Name System (DNS), and the X11 (dominant windowing protocol for Unix). They are being extended by the more automated operations of the World Wide Web (WWW) browsers, Mosaic and Netscape. They pose serious security problems. Any organization considering implementing a firewall must be aware of how to make these services available yet secure.

D.1. FTP

FTP is probably the most-used service, after E-mail, on the Internet today. It allows users to download files and enables organizations to make easily available a variety of documents, software, and graphic images. Using FTP presents various security headches. An FTP deamon normally runs with root privileges and doesn't give up this status. If an intruder manages to get into the system by making a hole in the FTP implementation, he can get into the system with maximum privileges. It was a known security hole in sendmail which is another privileged program. Also, FTP uses port 21 as the outgoing or control channel and port 20 as the incoming or data channel. This means that the fire door will be open whenever FTP is allowed. There are several ways to get a secure FTP connection through a firewall. Some of these different ways are as follows [4] :

1. Pass FTP requests to a proxy server which is a restricted cut-down version of FTP that's

known to be secure and will only permit what is allowed.

2. Modify the FTP software so that it will talk to only a restricted range of ports, for which firewall filtering or screening rules are in place.

3. Use the passive option, if it's available, to indicate that the remote FTP server should permit the client to initiate connections. This will work, however, only when the remote FTP server supports that operation.

4. Finally, it is possible to build client versions of FTP that are linked against a SOCKS library (a generic proxy implementation).

D.2. DNS

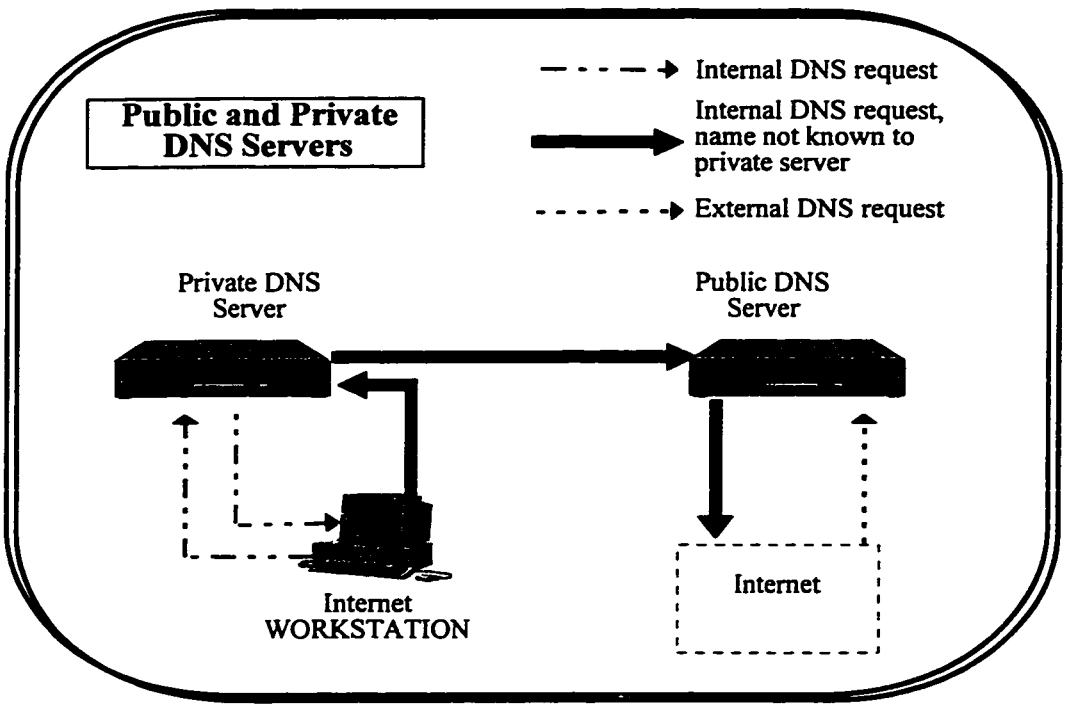
DNS is vital to the correct operation of the internal mail and other functions. It is just like the telephone book that lets you know who's who and where in the network. But its structured database contains important and often sensitive information about the system that we don't want to expose to the outside world. Therefore, many organizations don't want to make their host names public. In some cases, establishing an Internet connection may require reconfiguration of the internal networks and domains. However, there are workarounds, one approach works by redirecting DNS clients so that they talk to a DNS server that is on a different machine [4] :

1. First, on the outer side of the firewall host, we set up a DNS server that the outside world can talk to. To the outside world, this is what the domains look like, although, in fact,

they are seeing a restricted set of names (and aliases) that tells them only what we want them to know. This can be called the public server.

2. Next, we set up another DNS server on an internal machine. This server is in fact the real thing, and it contains information about the hosts. We make sure that this server forwards any queries it can't resolve to the public server.

3. Finally, we set up our DNS clients, including any clients on the machine that hosts the public server, and use the internal server. This is the key.



DNS

Internal client makes a request and gets an answer from the internal server or when asking about an external host, from the Internet but relayed through the public DNS server.

External requests, however, are handled completely by the public server with its restricted information as shown in the above Figure (DNS).

This technique can be an effective way for an organization to quickly set up a restricted DNS gateway without having to conduct significant internal reconfiguration. But, hiding names in the DNS doesn't stop host names from leaking out in mail headers, signature files, and so on.

D.3. X11

X11 deals with the user's terminal as a server. This philosophy provides many benefits, but, it also puts the system at risk. Applications connected to an X11 server have the power to seriously compromise security. Spoofing problem still hasn't gone away. According to Computer Emergency Response Team (CERT), while some routers cannot be programmed to defeat such attacks, others can. Even though these are not X11-specific, they illustrate the dangers of spoofing. Remote systems that can gain or spoof access to a workstation's X11 display, can monitor keystrokes that a user enters, download screen dumps that contain sensitive data, generate commands that appear to originate at the keyboard, and so on. Because of these problems, most firewalls block all X11 traffic unless they have a specially written application proxy to handle it.

APPENDIX E
Common Security Practices

In the 1994 year-end report by he Ernst & Young auditing and management consulting firm, to assess the current state of security practice, 1271 information security managers were surveyed. According to the report, about half or more of those companies running mission-critical systems on LANs believe their security is unsatisfactory.

- The biggest concerns are:

1. %85 for network security.
2. %83 for unauthorized external access.

- In response to a different question, expressed concerns were:

1. %93 about the unavailability of network service.
2. Followed by the fear of interference with operations and loss of message confidentiality or integrity.
3. %83 about the inability to identify network users.
4. Over %50 faced losses or interruptions in the past two years.

- Connectivity for organizations with over 2500 employees, shows that:

1. %55 say their networks are accessed by customers.

2. %46 by suppliers.

3. %33 by both.

4. %45 use the Internet or other public data networks.

5. %88 use E-mail.

6. Internally, most LANs and departmental mini computerss are connected to a central computing resource.

- The bigger the system, the safer it is as indicated by the study and shown below:

1. %4 of MVS mainframe users believed software security was inadequate.

2. %22 For Unix machines.

3. %14 to %19 range for LANs, with NetWare at the top.

- Desktop machines were considered the least secure with:

1. MS-DOS computers reported as %57.

2. Macintosh computers reported as %47.

3. Windows did better, at %37.

4. OS/2 systems %27, almost like the Unix boxes.

- An outline of the extent to which security managers are using control measures for the above concerns is listed below:

1. Antivirus software (%91).
2. Dial-back or secure modems (%54).
3. Firewalls (%45).
4. File encryption (%36).
5. PC hardware security devices (%33).
6. Telecommunications encryption (%22).
7. Message authentication coding (%17).

REFERENCES

[1] Hanafy Meleis. Toward the information network. *IEEE Computer Magazine*, pages 59–67, October 1996.

[2] B. Guha and B. Mukherjee. Network security via reverse engineering of TCP code: vulnerability analysis and proposed solutions. *Proceedings of IEEE INFOCOM '96. Conference on Computer Communications, San Francisco, CA, USA*, 2:603–610, March 1996.

[3] S. Cobb. Establishing firewall policy. *Southcon/96 Conference Record, Orlando, FL, USA*, pages 198–205, June 1996.

[4] Peter Stephenson. A blueprint for firewalls. *LAN Magazine*, 10(2):63–70, February 1995.

[5] Marcus J. Ranum and Frederick M. Avolio. A toolkit and methods for internet firewalls. *http:// www.tis.com/ docs/ products/ gauntlet/ Usenix.html.*

[6] B. Gassman. Internet security, and firewalls protection on the internet. *Professional Program Proceedings. ELECTRO '96, Somerset, NJ, USA*, pages 93–107, May 1996.

[7] David j. Stang and Sylvia Moon. *Network World Network Security SECRETS.* IDG Books Worldwide, Inc., 1993.

[8] S. Albert, V. A. Ashby, and S. E. Hicks. Reference model for data management security and privacy. *SIG Security Audit and Control Review, Spring/Summer*, 10(2), 1992.

[9] George Wang Weijun. Inter-networking security. *Singapore ICCS*, pages 1190–1194, November 1994.

[10] William F. Jolitz and Lynne Greer Jolitz. Role-based network security: network security at the operating-system level. *Dr. Dobb's Journal*, 20(5):80–83, May 1995.

[11] William Stallings. *Network And Internetwork Security: principles and practice.* Prentice-Hall, Inc., 1995.

[12] Judith Silver. Firewalls are the net's first, but not only, line of defense. *Government Computer News*, 14(24):44–45, November 1995.

[13] Prabhu Ram and Douglas K. Rand. Satan: Double-edged sword. *Internet Kiosk*, pages 82–83, June 1995.

[14] Peter Morrissey. Firewalls. *Network Computing*, 7(2):54–62, February 1996.

[15] Edwin E. Mier. Filtering out unwanted data packets. *Communications Week*, (575):70, September 1995.

[16] S. M Bellovin and W. R. Cheswick. Network firewalls. *IEEE Communications Mag-*

azine, 32(9):50–57, September 1994.

[17] N. E. Hastings and P. A. McLean. TCP/IP spoofing fundamentals. *Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications, Scottsdale, AZ, USA*, pages 218–224, March 1996.

[18] Uyless Black. *TCP/IP & Related Protocols (Second Edition)*. McGraw-Hill Inc., 1994.

[19] Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall International, Inc., 1996.

[20] S. M. Bellovin. Security problems in the TCP/IP protocol suite. *Computer Communications Review*, 19(2):32–48, April 1989.

[21] Robert T. Morris. A weakness in the 4.2 bsd unix TCP/IP software. *Computing Science Technical Report No. 117, AT&T Bell Laboratories, Murray Hill, New Jersey*, February 1985.

[22] M. Weiser. Program slicing. *IEEE Transactions on Software Engineering*, SE-10:352–375, July 1984.

[23] Joe Paone. Cyberspace invaders: (internet firewalls and data security, includes related article on the satan (security administrator tool for analyzing networks) computer program). *INTERNETWORK*, 6(6):33–36, June 1995.

[24] Clinton Wilder and Jason Levitt. Cure or curse? ready or not, the satan security tool is coming april 5. (security administrator tool for analyzing networks data security software). *InformationWeek*, (521):14–16, Apr 3, 1995.

[25] Dan Farmer and Wietse Venema. Introduction. *http:// recycle.cebaf.gov/ ~doolitt/ satan/html/ docs/ intro.html*, June 19 1995.

[26] Dan Farmer and Wietse Venema. Tutorials - security problems. *http:// recycle.cebaf.gov/ ~doolitt/ satan/html/ vulnerability_tutorials.html*, June 19 1995.

[27] Dan Farmer and Wietse Venema. Satan frequently asked questions (faq). *http:// wzv.win.tue.nl/ satan/ demo/ docs/ FAQ.html*, April 10 1995.

[28] David Newman and Brent Melson. Can firewalls take the heat? *Data Communications*, 24(16):71–80, Nov. 21 1995.

[29] Steffen Stempel. IpAccess - an internet service access system for firewall installations. *IEEE Symposium of Network and Distributed System Security, CA, USA*, pages 31–41, February 1994.

[30] Ibrahim Al-Kaltham and Khalid Al-Tawil. Achieving network security with firewalls. *Proceedings of King Fahd University of Petroleum and Minerals, Saudi Computer Society, 15th National Computer Conference*, pages 371–386, 17-19 Nov 1997.

[31] Michael B. Greenwald, Sandeep K. Singhal, Jonathan R. Stone, and David R. Cheriton. Designing an academic firewall: Policy, practice, and experience with SURF. *Proceedings of Internet Society Symposium on Network and Distributed Systems Security, San*

Diego, CA, USA, pages 79–92, February 1996.

[32] Ping Lin and Lin Lin. Security in enterprise networking: A quick tour. *IEEE Communications Magazine*, 34(1):56–61, January 1996.

[33] William R. Cheswick and Steven M. Bellovin. *Firewalls and Internet Security Repelling the Wily Hacker*. Addison-Wesley publishing company, 1994.

[34] Brent Chapman. Network (in)security through IP packet filtering. *In USENIX Security Symposium III Proceedings*, pages 63–76, September 1992.

[35] Edwin E. Mier. Another brick in the firewall. *CommunicationsWeek*, (575):65–68, Sep 18, 1995.

[36] Steven Lamb. Firewalls not created equal. *Computing Canda*, 22(1):35, January 1996.

[37] Washington University Saint Louis. Ftp server daemon. *Available for FTP from wuarchive. wustl.edu.*

[38] Robert J. Melford. Internet security planning requires a watchful eye. *Digital News and Review*, 12(10):16–17, May 1995.

[39] Paul Merenbloom. Final steps for creating a firewall and guarding internet access. *InfoWorld*, 16(32):59–63, Aug 8, 1994.

[40] Bob Melford. How to build an internet firewall. *Digital News and Review*, 13(1):22, January 1996.

[41] Gary Kessler. Build great firewalls. *Network VAR*, 3(6):29–33, June 1995.

[42] D. Koblas and M. R. Koblas. Socks. *ftp:// ftp.nec.com/pub/ security/ socks.cstc/ socks_usenix_paper.ps.gz.*

[43] D. Brent Chapman and Elizabeth D. Zwicky. *Building Internet Firewalls*. O'Reilly and Associates, inc, 1995.

[44] Karanjit Siyan and Chris Hare. *Internet Firewalls and Network Security*. New Riders Publishing, 1995.

[45] Tecnologic Partners. Daemon logic. *Computer Letter*, 11(37):1–6, November 1995.

[46] Clifford Meth. Hardware solutions improve data security. *Electronic Design*, pages 53–56, May 1995.

[47] G. Winfield Treese and Alec Wolman. X through the firewall, and other application relays. *Proceedings of USENIX Summer Conference, 1993. Also available as Cambridge Research Lab Technical Report 93/10, Digital Equipment Corporation*, May 1993.

[48] George Wang Weijun. Inter-networking security. *SINGAPORE ICCS/94*, pages 1190–1194, November 1994.

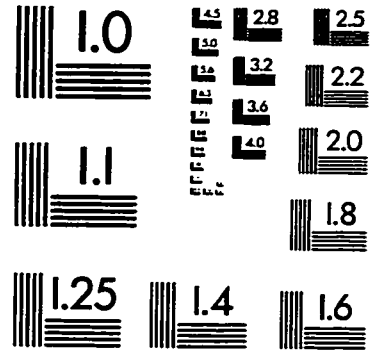
VITA

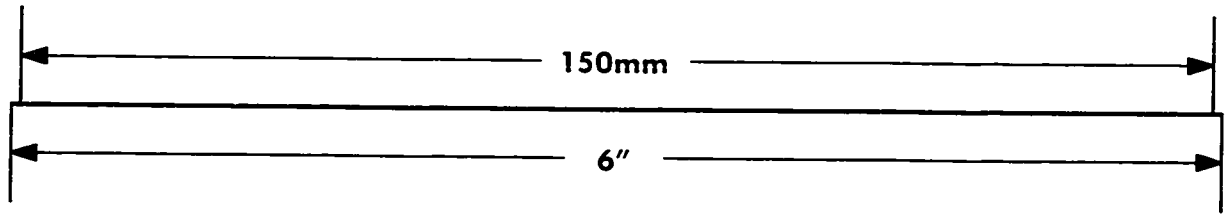
Ibrahim Abdul-Rahman Al-Kaltham, born at Al-Aflaj, Saudi Arabia in 1959.

Received Degree of Bachelor Science in Computer Science from the University of Tulsa at Tulsa, Oklahoma, United States of America, on May, 7, 1983.

Completed Master's degree requirements at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, on December 1997.

IMAGE EVALUATION TEST TARGET (QA-3)





APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989